



Welcome to JRiver Interact

Welcome, **Guest**. Please [login](#) or [register](#).
Did you miss your [activation email](#)?

Forever

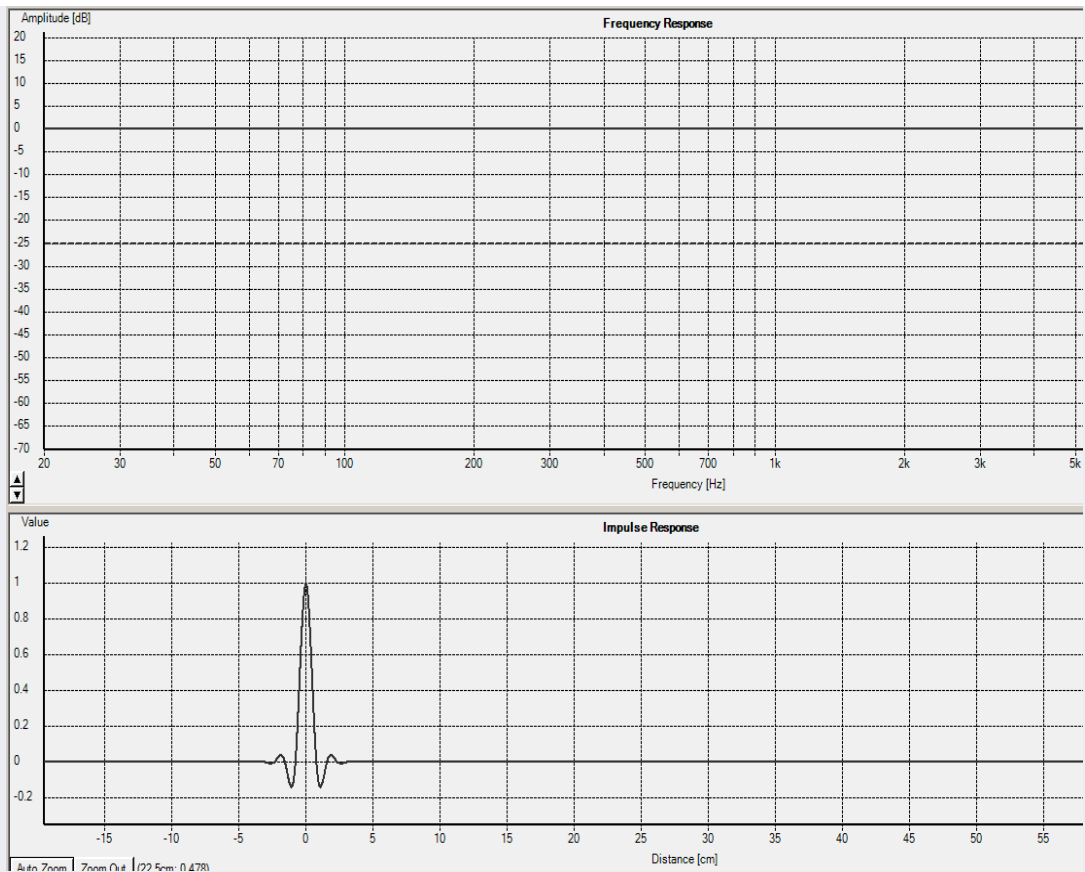
Login with username, password and session length

INTERACT FORUM > Devices > Sound Cards, DAC's, Receivers, Speakers, and Headphones (Moderators: [Forum Admin](#), [bob](#)) > **Guide to Speaker/Room Correction Using Freeware and JRiver**

[« previous](#) [next »](#)

Pages: [1] 2 3 [Go Down](#)

Author	Topic: Guide to Speaker/Room Correction Using Freeware and JRiver (Read 58623 times)
mwillems MC Beta Team Citizen of the Universe Posts: 3580 	<p>Guide to Speaker/Room Correction Using Freeware and JRiver « on: February 17, 2014, 12:03:21 pm »</p> <p>This is a step by step guide on how to do some basic speaker and room correction using JRiver and free software. This guide was inspired by MitchCo's guide to using Acourate for speaker and room correction http://www.computeraudiophile.com/content/529-acourate-digital-room-and-loudspeaker-correction-software-walkthrough/. My goal with the guide was to produce, using only free software and JRiver, a process that is "semi-automated," but still produces good results.</p> <p>I'll be using a home-made 2-way pair of bookshelf speakers as my example throughout. I built them ten years ago with speaker elements I bought at radio shack, and they were the first pair of speakers I ever built. So needless to say, knowing what I know now, they could benefit from some optimization.</p> <p>This guide should work fine for any bookshelf or full-range speaker with a built-in passive crossover (I plan to address subwoofers and active crossovers for bi-amping in a later guide). Different rooms and different speakers have different needs, so this is by no means the only way to do it, or even necessarily the best way to do it for your system. But it's intended to be a good place to start that will probably yield some improvements in an untuned system. My plan is to try and keep the guide relatively non-technical, but to provide enough background so folks can understand why I'm suggesting the steps I'm suggesting.</p> <p><u>What you'll need:</u></p> <ol style="list-style-type: none">1) JRiver Media Center version 18 or above, with the loopback feature configured and working correctly. If you need help getting loopback up and running, check out the following thread: http://yabb.jriver.com/interact/index.php?topic=70242.msg486195#msg486195. EDIT: If you have JRiver 20 or above you should be able to use the new WDM driver in place of loopback, just make sure it's set as the system default.2) A calibrated Microphone (with a calibration file) and a way to connect it to your computer (either a USB or phantom powered mic is fine)3) Holm Impulse (free measurement software): http://www.holmacoustics.com/holmimpulse.php4) Room EQ Wizard (free measurement and automated room correction software): http://www.hometheatershack.com/roomeq/5) RePhase (optional, free convolution filter designer): http://sourceforge.net/projects/rephase/6) Basic technical information about your speakers, usually found on their specification sheet or operator's manual (e.g. what frequency the drivers' crossover points are, where the port is tuned in a vented speaker, etc.)7) An afternoon. <p><u>Measurement Technique: Separating the Speaker from the Room.</u></p> <p>What we're measuring:</p> <p>Frequency response: If you fed a theoretically ideal loudspeaker a flat sine sweep it would have a frequency response that looked like the solid bar grey line at 0dB in the graph below:</p>



An ideal speaker would reproduce exactly what was fed to it without any non-linearities.

Phase: If you tracked the phase of that same theoretically ideal loudspeaker, it would look the dotted line in the top graph above.

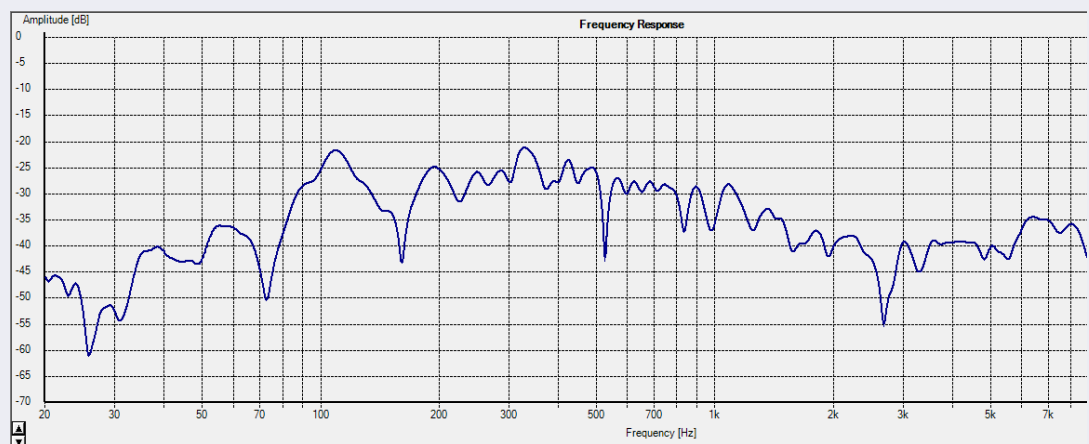
Impulse Response: If you looked at a band-limited impulse response (an impulse shows how the loudspeaker reacts to a sudden loud signal) of that same theoretically ideal loudspeaker, it would look like the trace on the bottom graph

These three measurements are by no means the only ways to measure a speaker's performance, but they're the ones we'll be using in this guide.

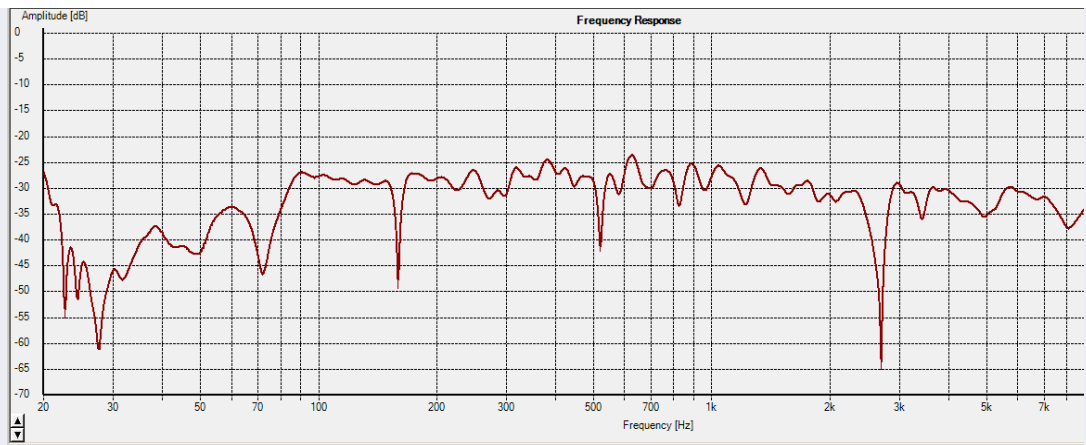
All real speakers will exhibit non-linearities in their frequency response and time-domain response (phase and impulse): no speakers are perfect. In addition, rooms will also introduce significant non-linearities, that will vary depending on the speaker position and the listening position in the room.

Anytime you attempt to measure a speaker in a room, you're really measuring the interaction of (at least) two systems: the loudspeaker and the room. It's very important, to the extent possible, to address the two systems separately. Why is that separation important? Because many loudspeaker non-linearities can be fruitfully corrected using EQ, but many room effects cannot effectively be corrected in EQ. I'll provide a quick example illustrating what happens if you don't address them separately:

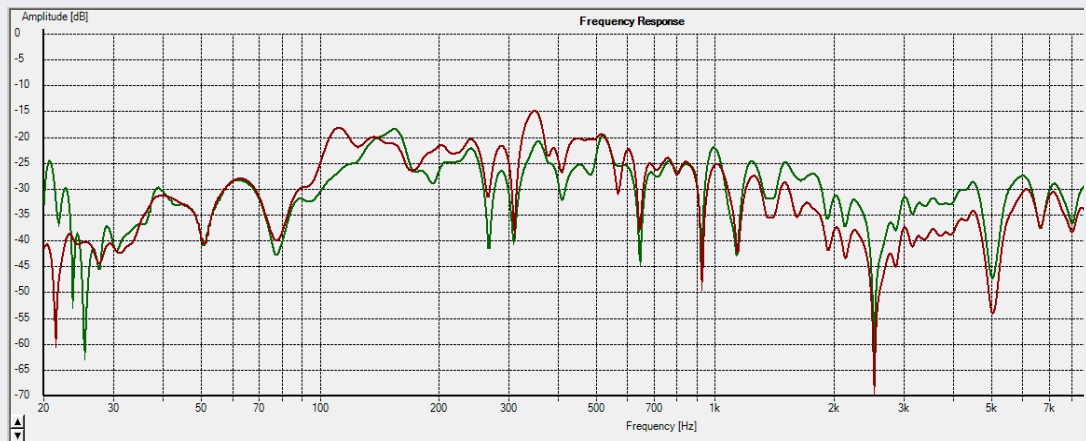
Here's my uncorrected speaker response at the primary listening position (room and speaker together):



Now I feed that response into an automated room correction suite (REW), and here are the results from the same position:



Looks much better, right? There are some things it didn't manage to fix (more on these "uncorrectables" in the EQ section below), but it looks much flatter across the band. Here's a measurement about a foot away from that position (the corrected trace is in green, with the uncorrected measurement at that position in red for comparison):



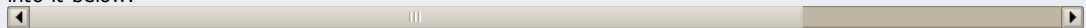
Some parts are improved by the automatic correction, but other parts are not; the correction is actually worse than the uncorrected trace in some positions. Attempts to strongly EQ a room's response a) will not always work, and b) even when they "work" will usually produce unintended consequences even a short distance away from the measurement position. With distances longer than a foot, expect even more unpredictable consequences.

By contrast, EQing the **loudspeaker's response** (as opposed to the **room's response**) can be done with much more consistent results, assuming you can first isolate it from the room.

Isolation Techniques:

Professionals measure speakers in an Anechoic Chamber or simulate an anechoic response by using an elevated platform outside (Altec engineers reportedly used a thirty foot platform for measurement). Most folks don't have access to an anechoic chamber (I certainly don't), and measurement outside isn't always an easy option (large speakers, apartment living, long cable runs, etc.). If you can easily and safely measure your speakers outdoors and reasonably high off the ground, it's worth doing, but in the interest of keeping this guide relevant to the largest number of people possible, I'll be focusing on in-room measurement through close-micing.

Close-micing is just what it sounds like: placing the mic very close to the speaker when measuring. Because the speaker's response will tend to dominate the response at short distances, you can get a quasi-isolated speaker response by taking measurements close to the speaker. There's an art to this, and I'll go into it below.



Logged

JRiver on Windows 10 64 bit || Arch Linux 64 bit with Gnome
Steinberg UR824 -> Homebrew Bi-Amped Mains ("Project Warhom")

Check out my Guides:

[Guide to Speaker Correction using JRiver](#)
[Guide to JRiver on the Raspberry Pi](#)
[Casting Audio to JRiver](#)

mwillems
MC Beta Team
Citizen of the Universe
Posts: 3580

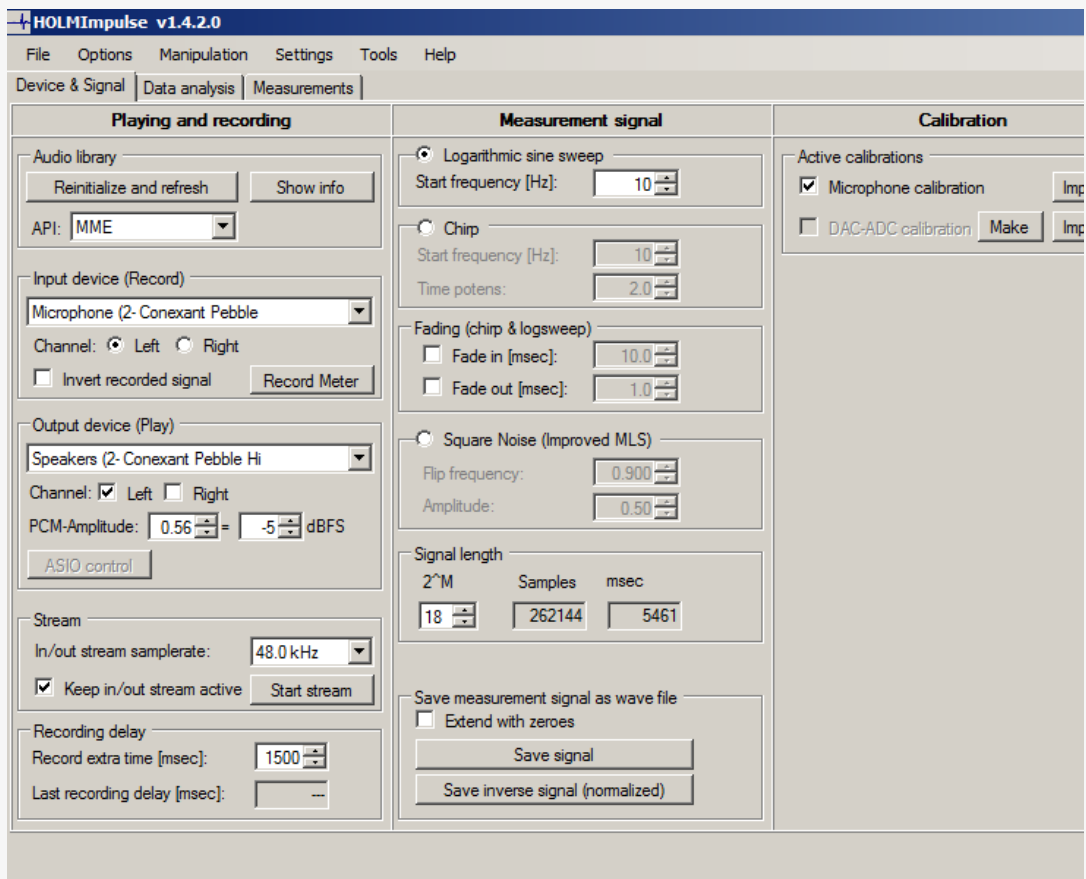


Re: Guide to Speaker/Room Correction Using Freeware and JRiver
« Reply #1 on: February 17, 2014, 12:03:32 pm »

Setting up Holm:

I recommend starting loopback in JRiver before starting up Holm. Holm will try to take exclusive control of sound outputs, which can cause errors in JRiver, but Holm usually behaves well if loopback is already started before you load it up. EDIT: JRiver's WDM driver greatly simplifies this issue, and I recommend using it in place of loopback if you have MC 20 or above.

There are quite a few settings in Holm, but I'll be focusing on a few important ones:

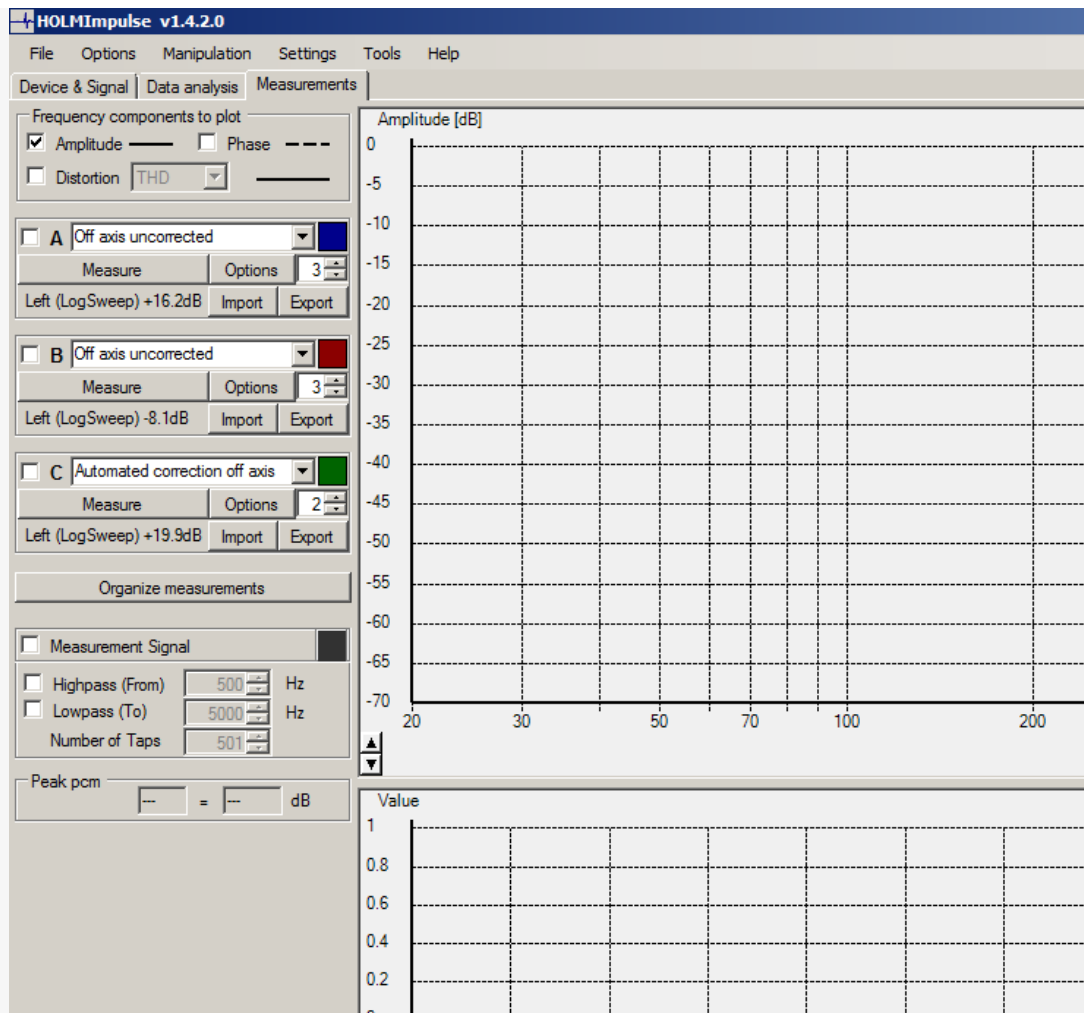


Under audio library you have a choice of whether Holm will be communicating with MME, direct sound, or ASIO. Because we'll be using loopback or the WDM driver to feed Holm's output to JRiver, I'd recommend sticking with direct sound until you're more comfortable with the program. Your output device should be the JRiver WDM driver or the windows default audio device that you're using for loopback (in my case the motherboard soundcard on my laptop). Beneath the output you can see which channels Holm will output to; I recommend only measuring one channel at a time to start. Your input should be your microphone.

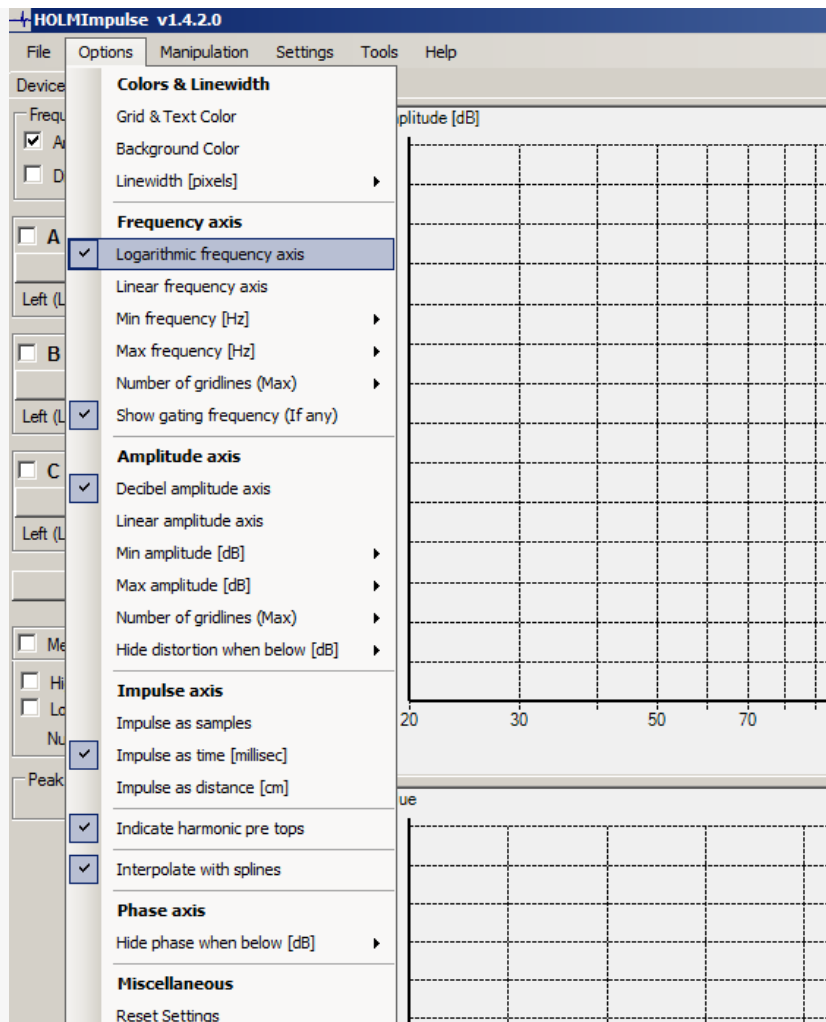
Under measurement signal, I recommend selecting "logarithmic sine sweep" and selecting a start frequency that is at least an octave lower than the lowest frequency you plan to test; by default it's set to 10 Hz, which will give you a more-than-full-range sine sweep. Most speakers can't reproduce tones that low, but this should be harmless at most normal volume levels with professionally designed speakers. If you're testing homebrew vented speakers that don't have a highpass to protect the woofer, you should start the sweep higher, and ditto if you're measuring unprotected tweeters, etc. If you have any doubts, make double sure to start measuring at a quiet volume and work up slowly.

Under mic calibration, you should import your mic's calibration file. If you're using a non-USB mic, you can also calibrate your DAC-ADC by connecting a cable from the output of your soundcard to the input you're using. If you're using a USB mic, there's no practical way to do this calibration, but the ADC built into the USB mic should be (to some extent) captured by the mic calibration file. The DAC-ADC calibration step is optional, in my opinion; the three DACs I've personally measured were all flat enough that there was no meaningful calibration necessary.

Now on to the Measurement Screen:



Here there are options to display frequency response, phase, harmonic distortion products, and impulses for up to three measurements at a time. Although you can only display three measurements at a time, Holm can hold (in working memory) up to fifty measurements at a time (accessed by hitting the drop down menu next to each measurement). I recommend giving each measurement you make a descriptive name as you go. A few more settings:



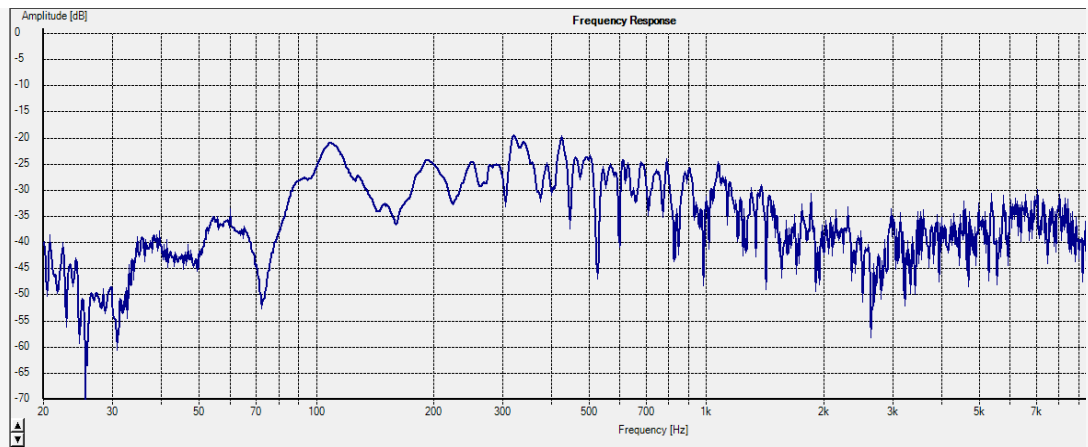
There are a lot of settings in the menus across the top (as shown above), and most are a matter of taste and convenience, but I strongly recommend making sure that "logarithmic frequency axis" is checked. I'd also recommend setting the min frequency to 20 Hz and the max frequency to 30,000 Hz

Measurement technique:

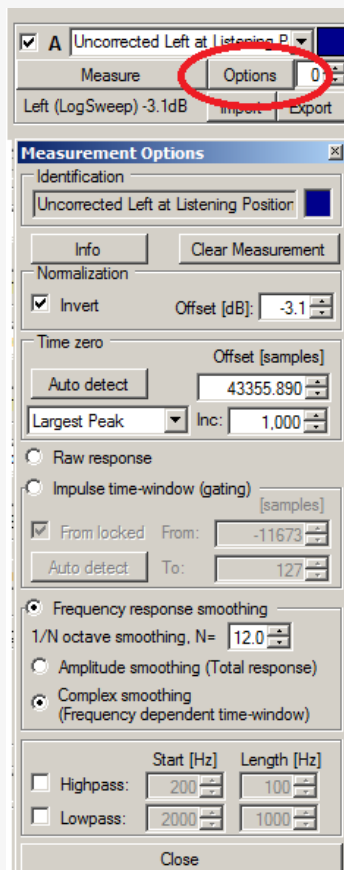
1) **Mic positioning** : Take note of what angle is specified for your mic's calibration file (0 degrees is common, so is 90 degrees). If there is no angle specified, it's usually safe to assume 0 degrees. A 0 degree calibration means that the mic should be pointed directly at the thing being measured in order for the calibration to be correct; likewise a 90 degree calibration means that the mic should be at a right angle to the thing being measured, etc.

2) **First measurement- Baseline** : Your first measurement should be at your primary listening position. This will not be a particularly useful measurement by itself, but will be the benchmark you'll be comparing with your final results to determine if you've made an improvement.

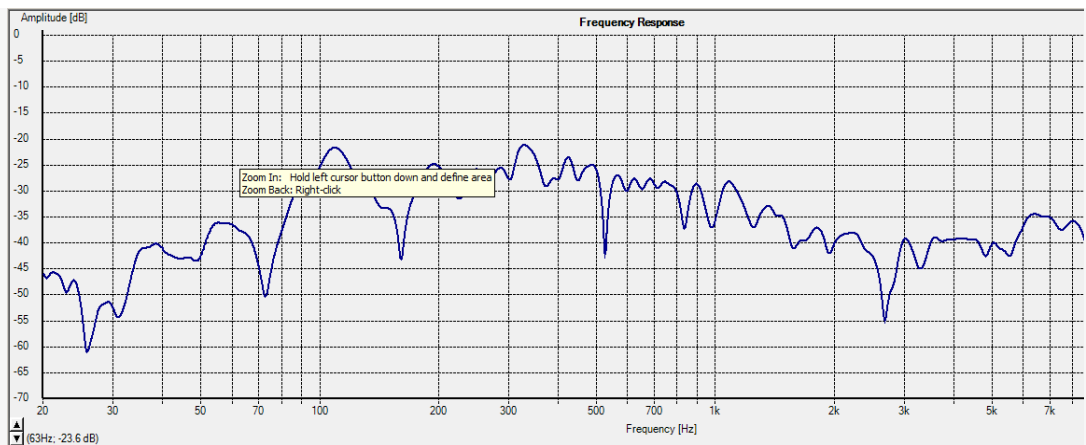
Place the microphone at ear height where you listen, and then set the volume in JRiver pretty low. Then hit the "measure" button in Holm. After that measurement completes, you'll want to keep adjusting the volume up in JRiver and running the measurement again until you're getting a reasonably loud sweep (about as loud as you'd normally ever want to listen to music). Once you get your volume set, do not change it during a measurement session (you want all the measurements to be measured at the same level for comparison purposes). Occasionally, you will hear a pop or crackle during the sweep, which will mess up your measurement. If you get one of those, just run the sweep again and make sure that it sounds smooth all the way through. Once you've gotten a reasonably loud successful measurement, you'll probably see something disturbing like this:



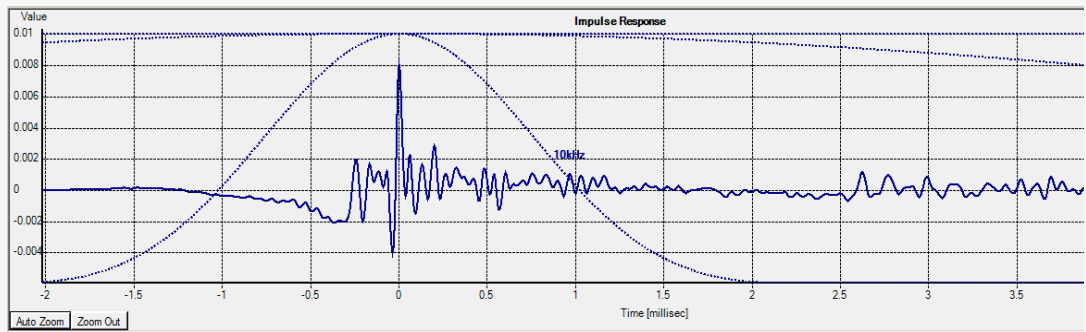
This is the raw response and includes all of the echoes and reflections from the room in the measurement. Open up the measurement "options" menu, and select "frequency response smoothing." Then select "Complex smoothing (frequency dependent time window)" and set the "N" to 12, as shown:



Frequency dependent time-windowing filters the response in a way that removes some of the room echoes and approximates how we actually hear sound better than the raw response does (our ears "ignore" certain late echoes). I recommend configuring all your measurements to use frequency dependent windowing during your first few measurement sessions, so that they're comparable (you may need to set these filter settings for each measurement, as you make it). You can play around with gating and different smoothing settings later on. One nice thing about Holm is that it stores the underlying raw measurement data, so you can change how it's displayed after the fact without making new measurements. After applying the filters, it looks like this:

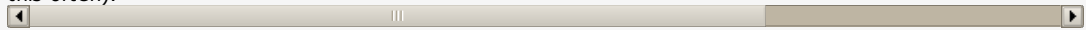


Second, auto zoom the Impulse response (or manually zoom in so that you're just looking at the first five or six milliseconds) and look at where the peak is. The largest peak should be a) on top and b) at the time zero as illustrated:



If the largest peak is not on top at the time zero, go to the options menu shown above, and go under "time zero" and select "largest peak" and click "autodetect." This should center the impulse on the time zero. If all the impulses you measure are upside down, you may need to check the "invert" box; if only some measurements are upside down, leave them be. It's important to get your time-zero lined up or your phase results will be wonky. You should check on this for every measurement (the "auto" usually gets it right, but it's worth confirming).

You should then repeat this first measurement with the other speaker, and then save your measurements (do this often).



Logged

JRiver on Windows 10 64 bit || Arch Linux 64 bit with Gnome
Steinberg UR824 -> Homebrew Bi-Amped Mains ("Project Warhorn")

Check out my Guides:
[Guide to Speaker Correction using JRiver](#)
[Guide to JRiver on the Raspberry Pi](#)
[Casting Audio to JRiver](#)

mwillems
MC Beta Team
Citizen of the Universe
Posts: 3580



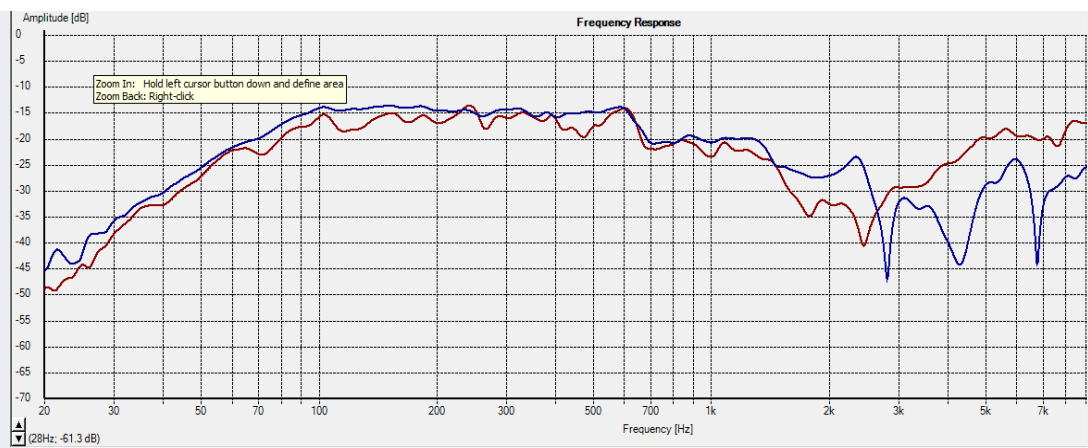
"Linux Merit Badge"
Recipient

Re: [Guide to Speaker/Room Correction Using Freeware and JRiver](#)
« Reply #2 on: February 17, 2014, 12:03:41 pm »

Now you'll take a series of measurements, as follows:

I recommend making five measurements for a two-way speaker: 1) on axis with the woofer from three inches away, 2) on axis with the tweeter from three inches away, 3) on axis with the woofer from one foot away, 4) on axis with the tweeter from one foot away, and finally 5) on axis with the tweeter from one meter away.

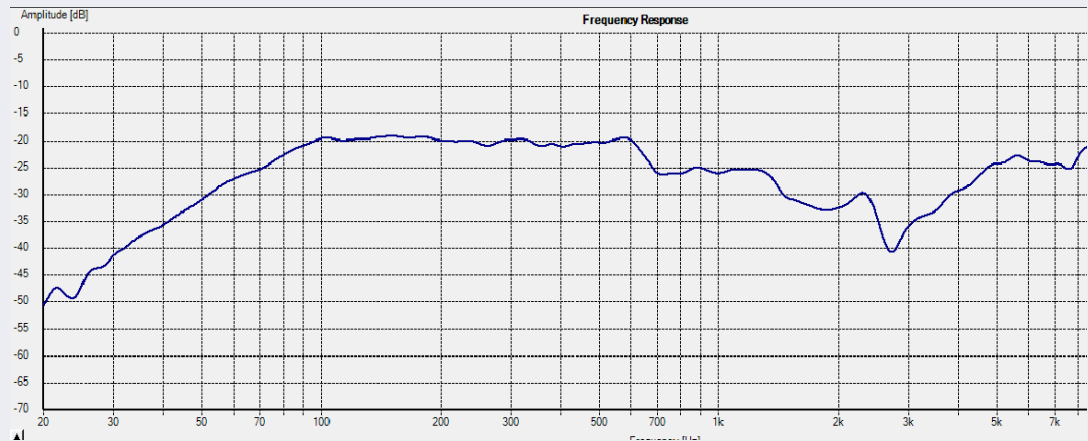
When I say "on axis" I mean that the tip of the mic should be centered horizontally and vertically on the speaker at the recommended distance. If you're using a three way or four way speaker, you should make additional measurements at three inches and one foot for each driver. Here are my two three-inch measurements overlaid (woof in blue, tweet in red):



You'll need to "stitch" the two measurements together to get a composite of the whole speaker's response. Make sure your "C" measurement slot is empty and then select manipulation-->stitching. For the frequency parameter, choose the crossover frequency from your speaker's data sheet (it should be relatively obvious from the phase measurements, but in case it's not, the tech specs will tell you), and choose a width that is about an octave wide at that frequency. In my example, the crossover is at 2500 Hz, and my width is about 2000 Hz. You should uncheck the box to "match phase before stitching."

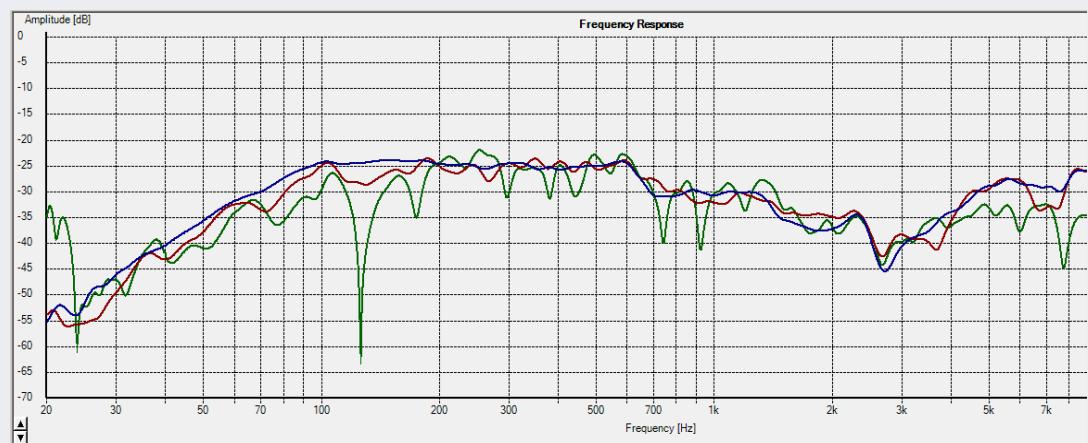
If the two measurements don't overlay closely, you should use Holm's Match function (a separate tool under the manipulation menu tree) **before** stitching. Match will adjust the level of the A-slot measurement to match the level of the B-slot measurement based on a specified frequency and bandwidth. As with the stitch, I'd recommend entering your speaker's crossover point as the frequency and entering a bandwidth that's about an octave wide at that frequency (i.e. about 750 Hz wide for a 1KHz crossover, etc.).

The result of the stitch looks like this:



You should do the same "stitching" process to the one foot measurements as well. You should now have three "final" measurements for your speaker: a composite three-inch measurement, a composite one-foot measurement, and a single one-meter measurement. Next you'll want to conduct those five measurements for the other speaker. You should now have three measurements for each speaker, which should look relatively similar.

Here are the three measurements overlaid for my left speaker: blue is 3 inches, red is 12 inches and green is 1 meter.

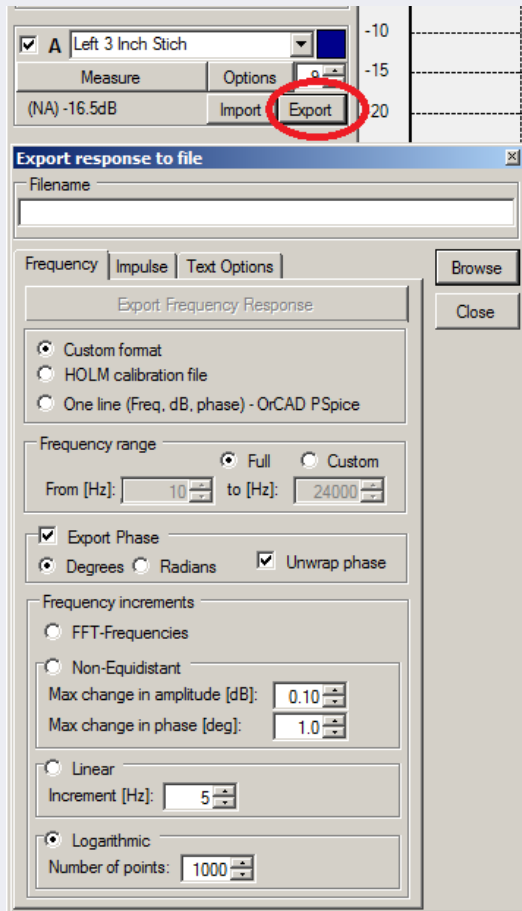


The three inch measurement shows the direct output of the driver with very little else (which can be deceptive, because the effect of the edges of the speaker cabinet itself are not always captured with a measurement that close). The one foot measurement shows the output of the driver along with some baffle diffraction (sound reflecting off the sides of the speaker cabinet) and a little bit of the room. The one meter measurement shows more of the room's effect, as you can no doubt see, as well as the interaction between the two drivers at ear height. The overlay of the measurements allows you to see where the areas most in

need of correction are:

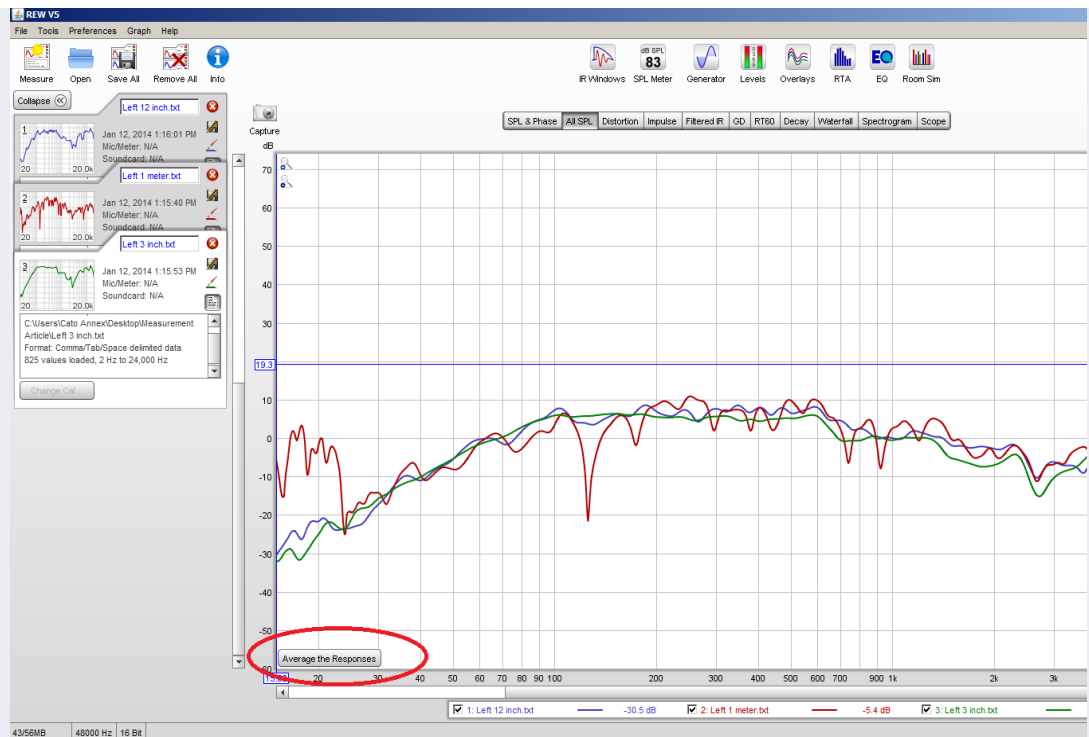
Frequency Response EQ

Now that you have your speaker measurements, what do you do with them? Unless you have an unusually "good" room, you've probably noticed that your three measurements are not identical, which makes the "right" EQ hard to sort out. I'm going to outline an automated method that should get you 90% of the way there. Step one is to open up the "export" menu in Holm

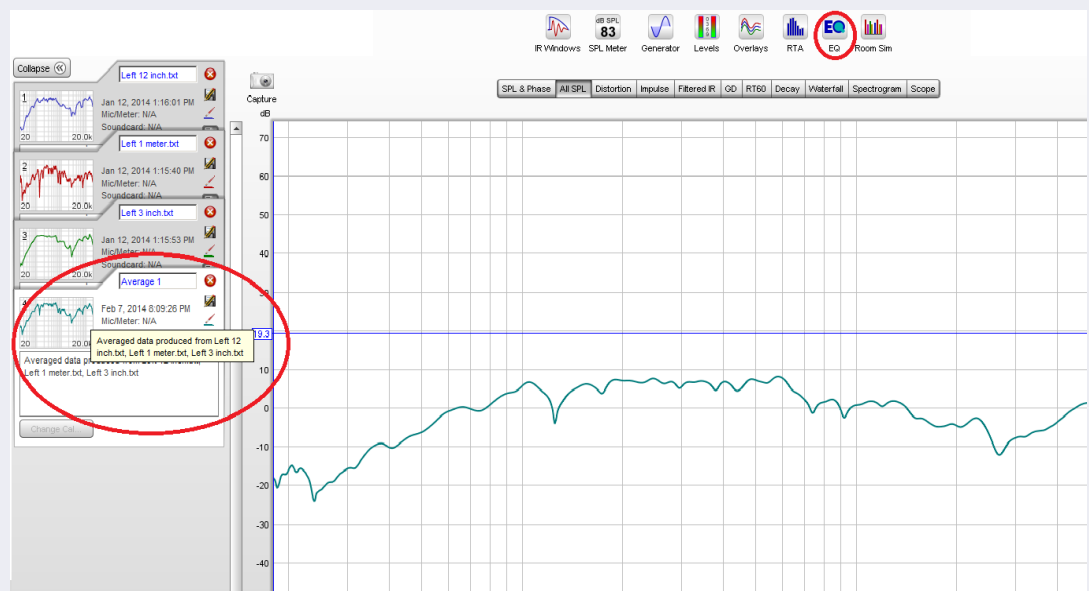


You'll want to export the frequency response of each of your three measurements for each speaker (not the impulse). Make sure you have frequency dependent windowing on and configured correctly for all of the measurements, as the frequency response will be exported as displayed, and you want the exported response to reflect frequency dependent windowing (this is important).

Once you have exported all the measurements, open up Room EQ Wizard (REW). REW has a measurement suite, but it does not support frequency dependent windowing or certain other features that Holm supports, so we will not be doing any measurements in REW in this guide. Instead, select "import frequency response" from the file menu and import the three measurements for your first speaker in REW. Tell it not to do a C-weighting correction, unless your mic is C-weighted (most calibrated mics are not). Once you have the three measurements for your first speaker imported, click the "All SPL" button along the top. Make sure all three are displayed, and then click on the "average the responses" button.



This gives you a hybrid of the three measurements that will produce a relatively "safe" EQ profile. Now, make sure the averaged response is selected and click the "EQ" button in REW.



Open up the settings panels as shown below:

Equaliser: Generic

TMREQ

DSP1124P

FBQ2496

DCX2496

SMS-1

R-DES

DSP-30

USM 810

MinIDSP

MinIDSP-96k

ADA PEQ

XP2040

Generic

Target Settings

Speaker Type:

Bass Limited

Crossover:

12dB/octave

Cutoff (Hz):

80

LF Slope:

LF Cutoff (Hz):

LF Rise Start (Hz):

100

LF Rise End (Hz):

20

LF Rise Slope (dB/octave):

0.0

HF Fall Start (Hz):

1000

HF Fall Slope (dB/octave):

1.4

Target Level (dB):

75.0

Set Target Level

Filter Tasks

Match Range:

22

to

10,000

Hz

Individual Max Boost:

6

dB

Overall Max Boost:

6

dB

Flatness Target:

2

dB

Match Response to Target

Manual Optimisation Controls

Optimise Gains

Optimise Gains and Qs

Optimise Gains, Qs and Frequencies

Retrieve Filter Settings from Equaliser

Send Filter Settings to Equaliser

Reset Filters for Current Measurement

Choose "generic equalizer," set speaker type to "bass-limited," and set "low rise slope" to zero (if you have truly full range speakers, feel free to set the speaker type to full-range instead). I recommend setting HF fall slope to 1.4dB and leaving the start frequency at 1KHz. This part is a question of taste, but some kind of HF roll off is generally desirable. These settings will give you a 6 dB roll between 1KHz and 20KHz; if you prefer things to be flat all the way out, set the HF fall slope to 0.

Logged

JRiver on Windows 10 64 bit || Arch Linux 64 bit with Gnome
Steinberg UR824 -> Homebrew Bi-Amped Mains ("Project Warhorn")

Check out my Guides:
[Guide to Speaker Correction using JRiver](#)
[Guide to JRiver on the Raspberry Pi](#)
[Casting Audio to JRiver](#)

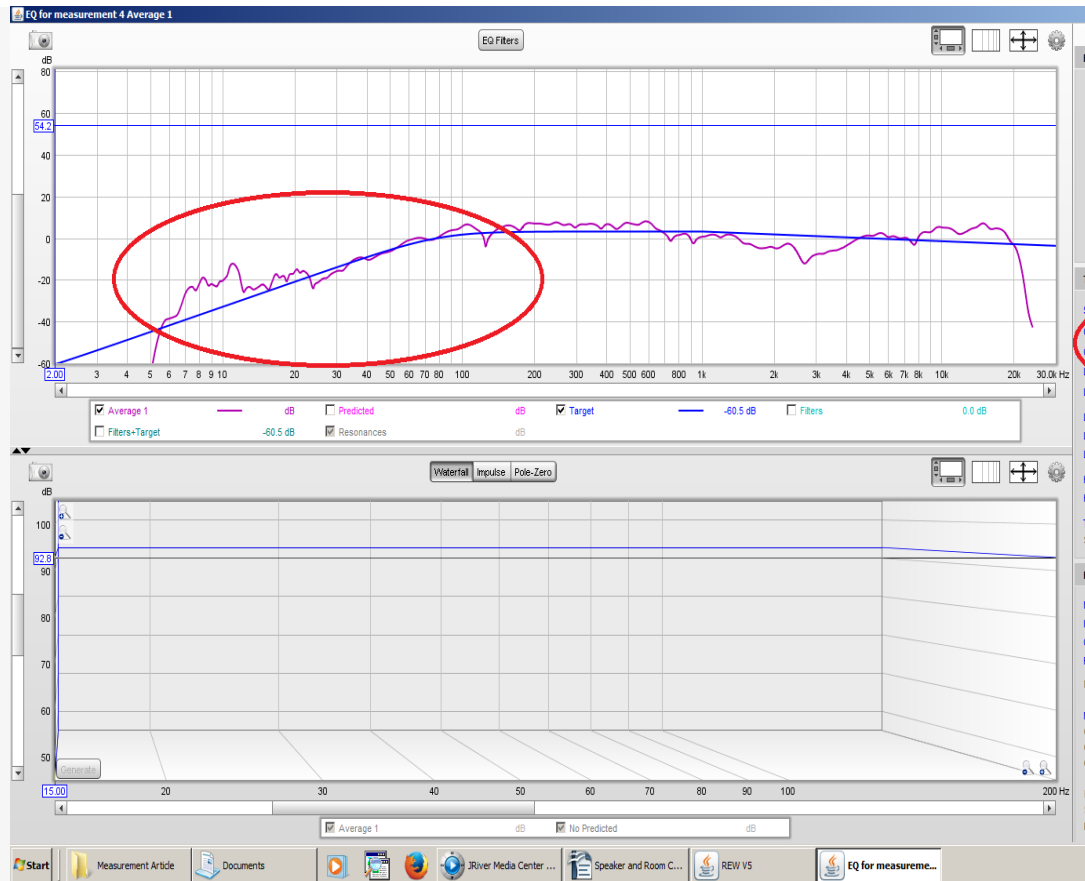
mwillems
MC Beta Team
Citizen of the Universe
Posts: 3580

Re: Guide to Speaker/Room Correction Using Freeware and JRiver
« Reply #3 on: February 17, 2014, 12:03:53 pm »

Next, click "set target level" to bring the target in line with your measurement (for some reason you may need to click it several times to get it to move all the way to the center of the measurement). Next pick the "cutoff frequency" and "crossover" that makes the cleanest line through the low frequency part of your measurement. You'll need to try a few settings, but when you're done it should look something like this:



"Linux Merit Badge"
Recipient



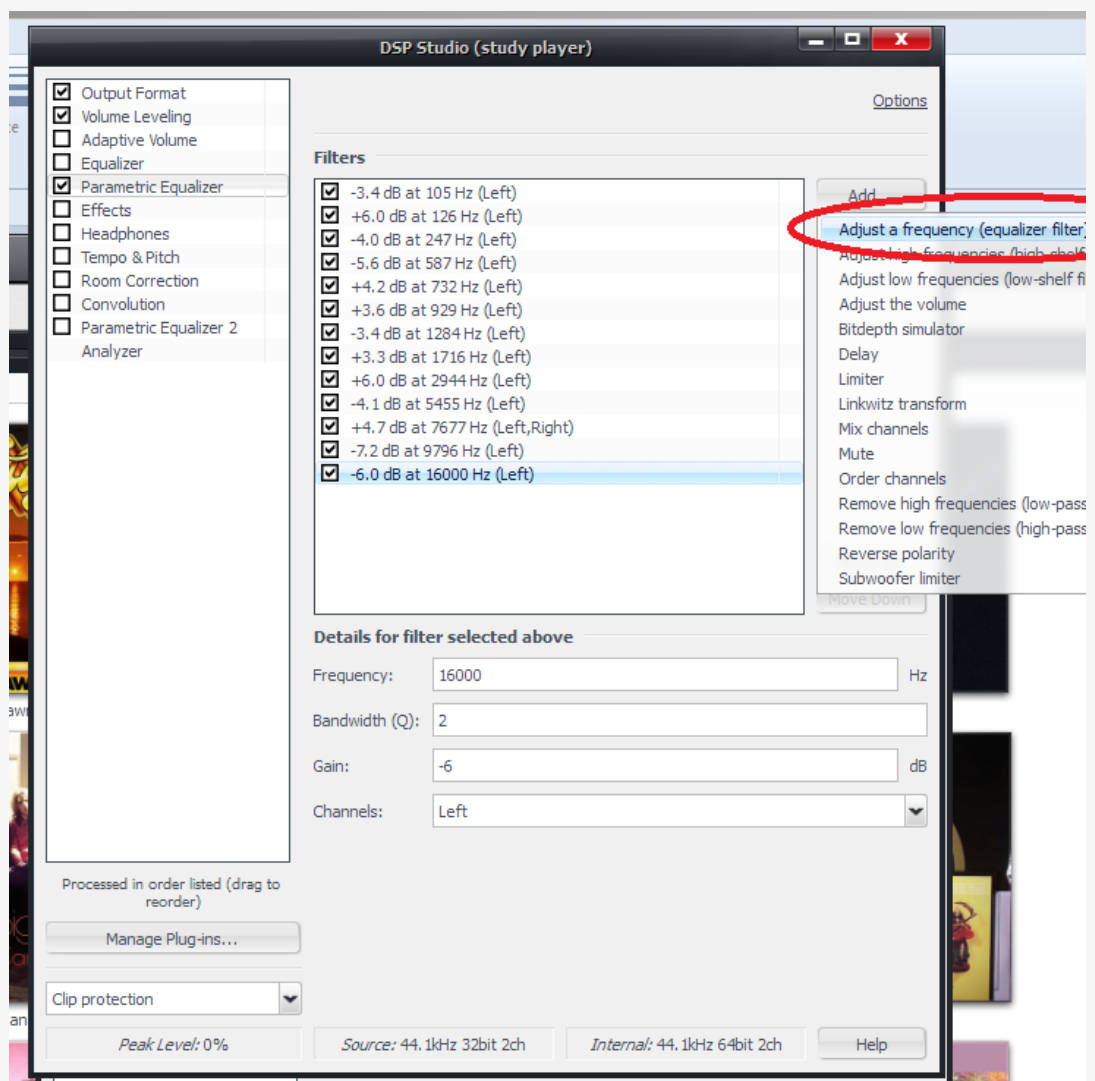
Set both maximum boost settings to 6 dB and the flatness target to 2 dB (feel free to set it to 1dB if your three measurements showed very close agreement).

For the "match range" I recommend setting the low end to the cutoff frequency from above and setting the top end to ~~10kHz~~ 20KHz (the latest beta version of REW now allows EQ to 20KHz, see below). If you set the match range lower and your low end target isn't a good fit, you could wind up trying to apply boost below the speakers terminal rolloff. If you try to EQ a sealed speaker flat below the terminal roll off without doing some math first you risk damaging your speaker and/or increasing your distortion levels dramatically (using eq to extend low bass response of a sealed speaker is an advanced topic). If you try to EQ below that point with vented speakers it not only might damage your speaker, it also won't work at all (the driver is unloaded below the port frequency and will just idly flap around). REW has some built in safety features to avoid this issue, but better safe than sorry: for your first time out set the bottom of the match range to the cutoff.

Once you've set all that up, click the "match response to target" button and watch REW work it's magic. Once it's done, click on "EQ Filters." You'll then see your filters in terms of frequency response, Q, and gain.

EQ Filters									
Average 1									
Sort Ascending by Freq									
Generic	Control	Type	Frequency	Gain	Q	Hz	Target T60	Mode T60	Filter T60
<input checked="" type="checkbox"/>	1	Auto	PK	105.5	-3.4	9.87	10.7	250	169
<input checked="" type="checkbox"/>	2	Auto	PK	126.0	6.0	13.79	9.1	170	340
<input checked="" type="checkbox"/>	3	Auto	PK	247.0	-4.0	1.00	247	11	7
<input checked="" type="checkbox"/>	4	Auto	PK	587.0	-5.6	2.09	281	11	6
<input checked="" type="checkbox"/>	5	Auto	PK	732.0	4.2	11.37	64.4	27	44
<input checked="" type="checkbox"/>	6	Auto	PK	929.0	3.6	1.00	929	2	3
<input checked="" type="checkbox"/>	7	Auto	PK	1,284	-3.4	3.74	343	8	5
<input checked="" type="checkbox"/>	8	Auto	PK	1,716	3.3	2.16	794	2	3
<input checked="" type="checkbox"/>	9	Auto	PK	2,944	6.0	1.15	2.56k	1	1
<input checked="" type="checkbox"/>	10	Auto	PK	5,455	-4.1	3.25	1.68k	2	1
<input checked="" type="checkbox"/>	11	Auto	PK	7,677	4.7	20.85	368	5	9
<input checked="" type="checkbox"/>	12	Auto	PK	9,796	-7.2	2.66	3.68k	1	1
<input checked="" type="checkbox"/>	13	Auto	None						
<input checked="" type="checkbox"/>	14	Auto	None						
<input checked="" type="checkbox"/>	15	Auto	None						
<input checked="" type="checkbox"/>	16	Auto	None						
<input checked="" type="checkbox"/>	17	Auto	None						
<input checked="" type="checkbox"/>	18	Auto	None						
<input checked="" type="checkbox"/>	19	Auto	None						
<input checked="" type="checkbox"/>	20	Auto	None						

My recommendation is that you write these down, and then copy them into JRiver's parametric EQ module, as shown below (it takes a minute, but has several serious advantages over the other method):



Alternatively, you can also export them as a WAV file which can be fed into JRiver's convolution engine, but this adds significant latency, which limits the usefulness of the filters (harder to use with applications that

aren't audio only, e.g. streaming video or video games). It will also prevent you from using convolution filters for phase correction in the next step. For those reasons, I don't recommended it, but if you do want to go this route, see the discussion of convolution filters below in the **phase correction** section of the guide for some instructions on how to use the exported WAV files in JRiver.

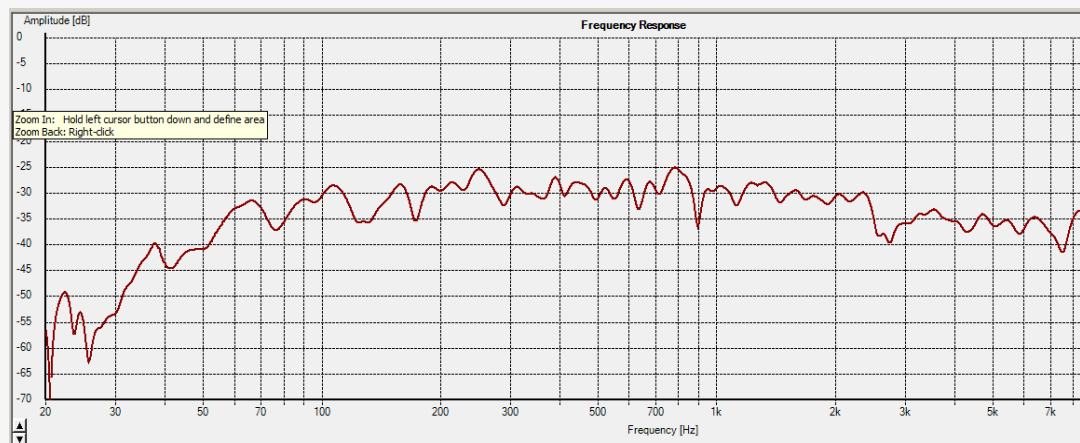
A few things about REW to be aware of:

- 1) EDIT: The latest beta version of REW now allows EQing all the way to 20KHz, so if you have the latest version, feel free to ignore this sub-paragraph ~~REW won't EQ above 10KHz, so if you have irregularities above 10KHz, you'll need to fix those manually by feeding in filters in PEQ and remeasuring. You may find it useful to run a Real Time Analyzer (RTA) while making the changes to speed things up. Here's a link to a guide mojavewrote about using REW as an RTA with JRiver: <http://yabb.jriver.com/interact/index.php?topic=69725.0>~~
- 2) REW won't EQ at the bottom of the range when the measurement dips below the target and doesn't rise above it again. This is designed to avoid the issues with speaker protection I mentioned above, but if your speaker has dips immediately preceding that roll off REW may ignore them. This can be worked around, but generally shouldn't be an issue if your cutoff settings are good.
- 3) REW will only generate up to 20 filters, but will generate fewer if it doesn't need all 20. If REW generates 20 filters for your response it probably actually needs more than 20, and you might want to consider generating the filters in two batches (bottom of the range to 600KHz, 601KHz to 10KHz, etc.)

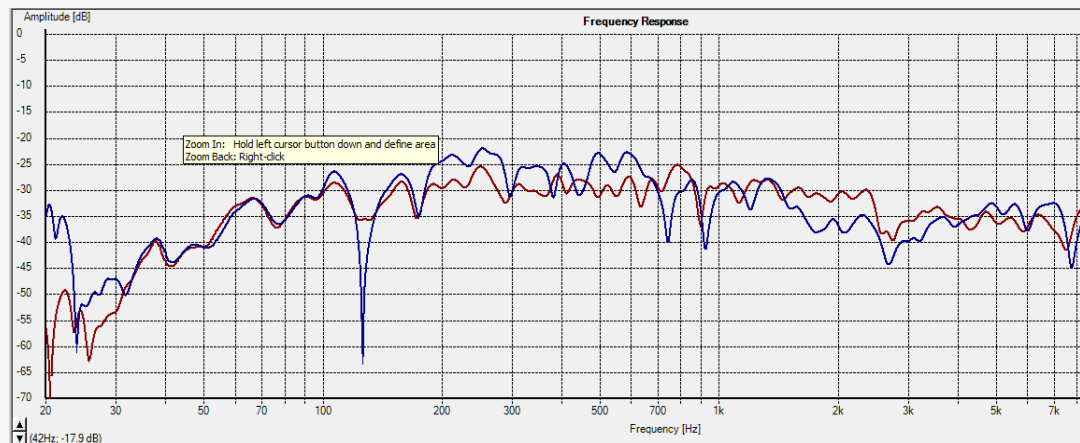
Once you've fed in the EQ filters into JRiver one way or another, go ahead and make any additional adjustments you may need to make. Lastly, add an **adjust volume** filter with the gain set to -6dB. This is to offset any gain added by the other filters you added above, which will prevent clipping. Depending on how you have JRiver configured, you may not need this filter, but if in doubt include the -6dB adjustment.

Once that's done, measure the corrected speaker again at ear/tweeter height at 1 meter (and save the results).

Here are my results:

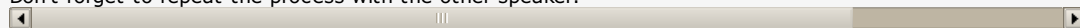


As you can see, much improved! Whereas the uncorrected measurement at 1 meter had peaks and troughs that were +/- 10dB between 100Hz and 20KHz, the corrected response is +/- 5dB around the target across the same band. Here they are overlaid so you can see the results more clearly (red is corrected, blue is uncorrected).



But you can also see that some elements that REW tried to correct were not fully corrected. Some things cannot be corrected by EQ; no amount of EQ will resolve them. They can represent a variety of room or speaker features, for example: standing waves, rear wall bounce cancellation, or in this case, the dip at 2.5KHz, which is the result of a suboptimal passive crossover. The first two can be potentially addressed through room treatment; the last can only truly be addressed by building a new crossover for the speaker or by bi-amping it (although it can be improved somewhat through phase manipulation, see below). As far as EQ is concerned, the best thing to do about uncorrectables is to avoid trying to EQ them at all, as you won't improve the response in the affected locations, and will potentially ruin the response everywhere else.

Don't forget to repeat the process with the other speaker.



Logged

Check out my Guides:

[Guide to Speaker Correction using JRiver](#)

[Guide to JRiver on the Raspberry Pi](#)

[Casting Audio to JRiver](#)

mwillems

MC Beta Team
Citizen of the Universe

Posts: 3580



"Linux Merit Badge"
Recipient

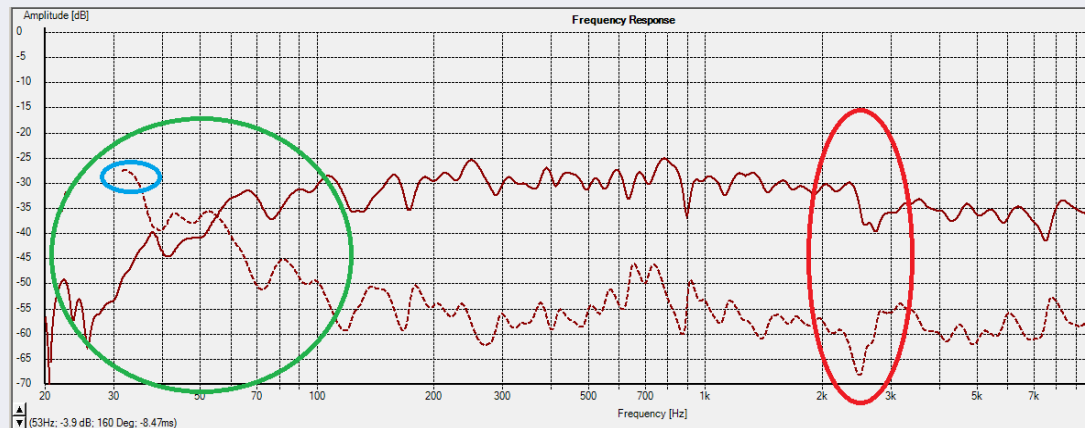
Re: Guide to Speaker/Room Correction Using Freeware and JRiver

« Reply #4 on: February 17, 2014, 12:04:02 pm »

Now that we've adjusted nonlinearities of the frequency response of the speakers, the next step would be to adjust the phase nonlinearities. Unfortunately, this is where the "semi-automated" part of the guide kind of runs out of gas: I'm not aware of a good, free, automated phase linearization solution that allows one to effectively adjust phase independently of frequency response. So for phase, we're stuck with manual tweaking, but I'll try to give you some general guideposts to navigate by. For these reasons, the next section is definitely **optional**. Hopefully, you've already made significant gains by this point, phase linearization is just the icing on the cake.

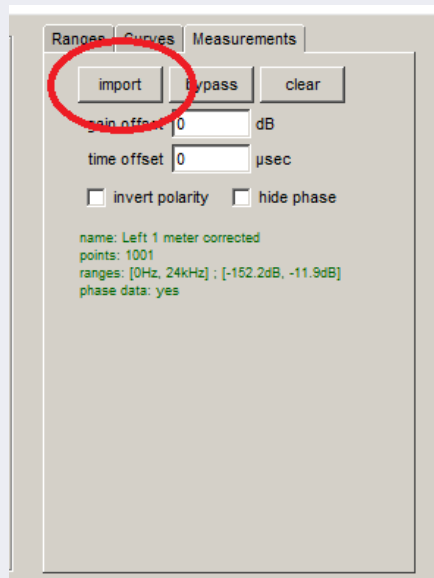
Speaker Phase Correction:

Now let's have a look at the phase of our measurements. Bring back up the last measurement you took (the corrected measurement). Here's the phase along with the frequency response.



You can see on this measurement (circled in red) where the crossover is, and where the sealed box begins to affect the phase at the low end (circled in green). A speaker with a passive crossover (unless it's a 1st order) will always have some kind of phase anomaly at the crossover point (usually more extreme looking than this one), and any speaker in an enclosure will exhibit some kind of phase anomalies at the low end (a vented speaker will have much greater phase rotation than a sealed speaker). You can also see that once the speaker output gets quiet enough the phase just stops or goes berserk (circled in blue). This is just what happens to a measurement when the frequency response falls off drastically; this can't/shouldn't be corrected.

To correct the phase, first export the frequency response of your corrected measurement in Holm the same way you did above, but make sure you have the "export phase" button checked. Then open up RePhase. Click on the measurements window tab, then click import:

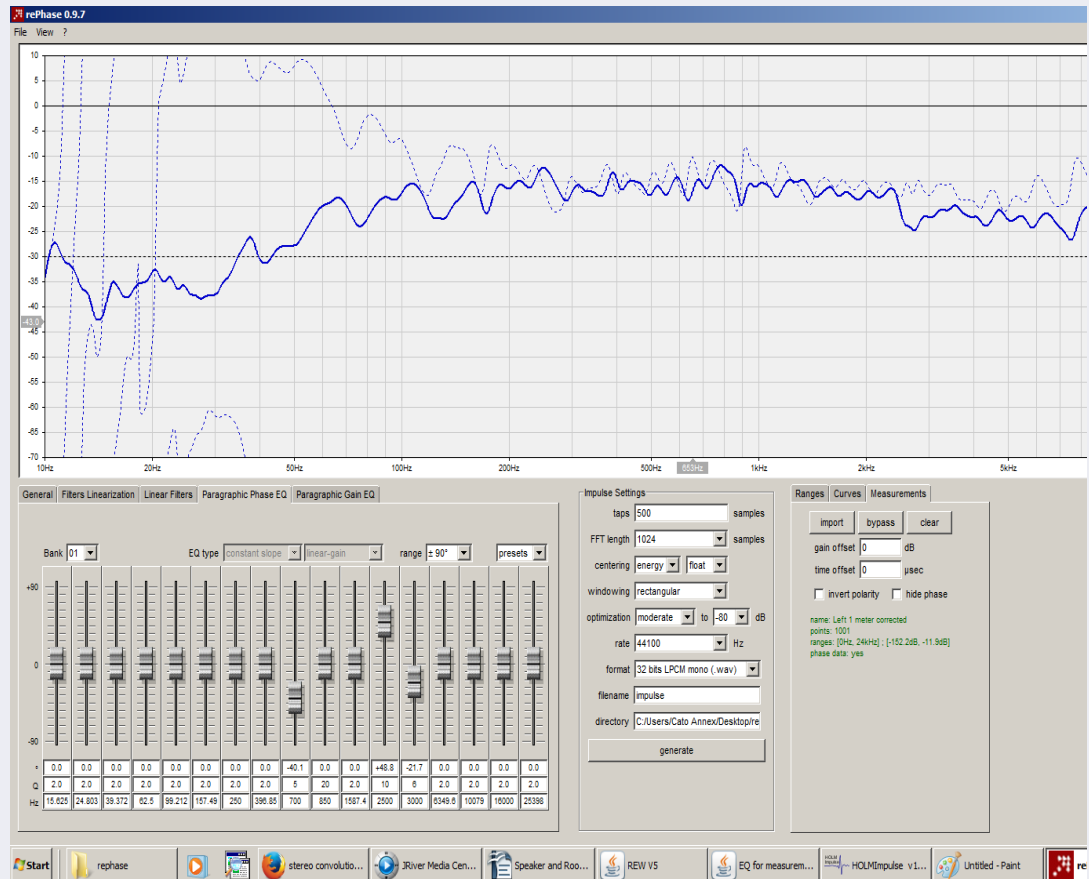


Now we can use RePhase to flatten the phase manually. Because phase distortion is much less audible than frequency response non-linearities, I'd generally advise taking it easy on phase correction and just work on correction of major phase wrap. The two main sources of phase wrap that I'd recommend correcting are crossovers and port phase distortion.

1) Crossover:

Under "filters linearization" RePhase has automatic pre-sets dialed in for linkwitz-riley crossovers. Your speaker may not have an LR filter, but I'd recommend fiddling around with the different linearization options to

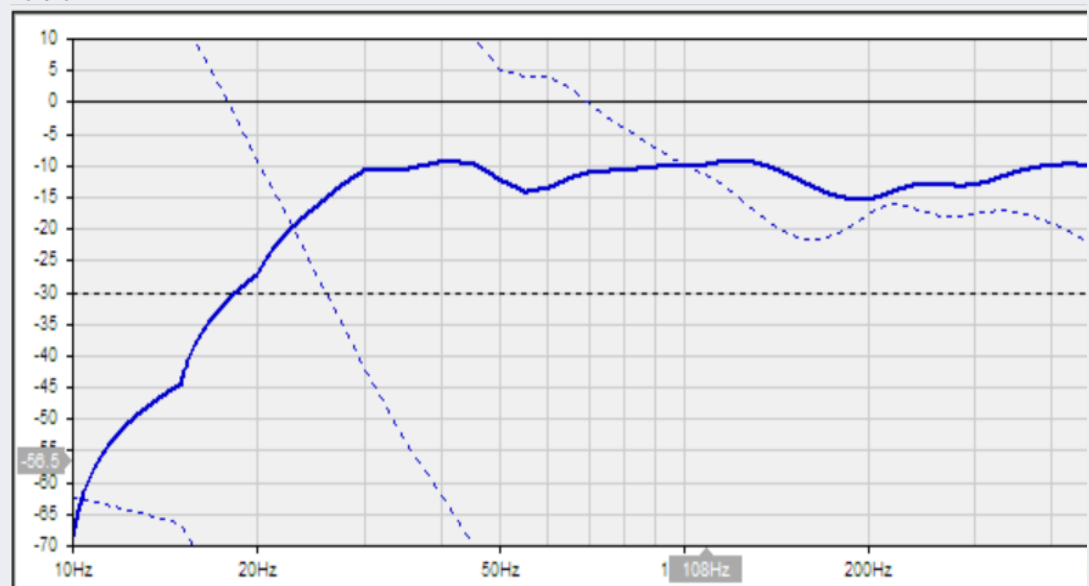
see if one of the presets partially or mostly addresses the phase wrap for your specific crossover. You can also use the paragraphic phase EQ to level it out (all the parameters on the bottom of the sliders are configurable). Just remember, higher Q means a narrower region of adjustment, lower Q means a wider region of adjustment. You can see how I used the paragraphic phase EQ to flatten out the phase in the crossover region below:



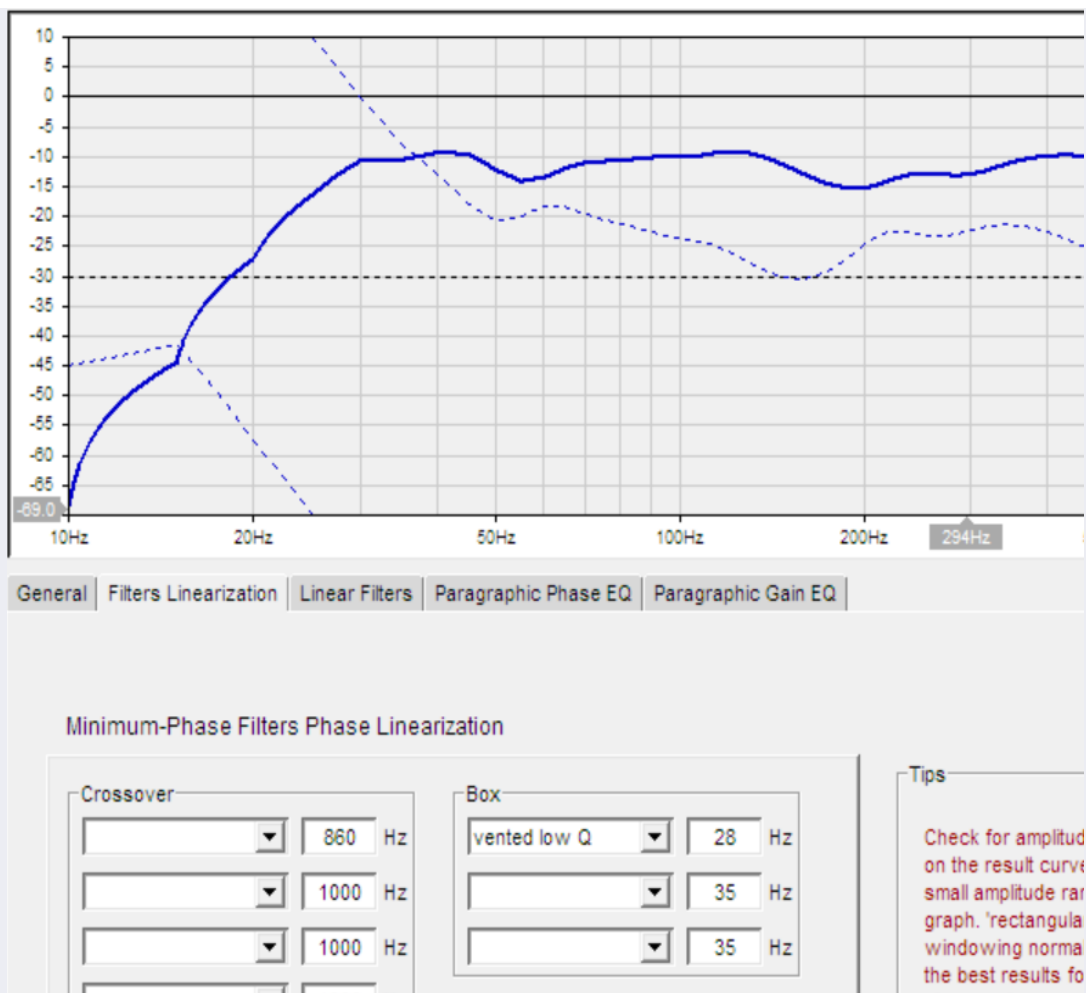
2) Port phase wrap:

Ports in vented speakers introduce phase wrap, distortion, and also ringing around the port frequency. Removing some of the phase wrap resulting from the port can not only improve the linearity of your phase response it can also reduce port ringing. Because my test speakers are sealed, I can't illustrate this on these speakers, but here's an illustration of un-wrapping the port phase rotation on a different speaker with a port tuned to 28Hz.

Before:



After:



To achieve this, I just used one of the "box" presets for a vented speaker (in this case a low Q speaker, because the box in question is very large). The illustrated speaker exhibited significantly improved time-decay response around the port frequency after being "linearized" in this way, and sounded much less "boomy." Time-decay measurements are a slightly more advanced topic, but I wanted to mention this potential advantage of linearizing some of the port phase wrap.

You don't necessarily want to flatten the low-end phase out completely, you just want the phase to look more like the wrap that a sealed box speaker would have. If your speaker is sealed, I wouldn't recommend linearizing the phase wrap at the bottom of the frequency range at all. Our ear expects phase shift as volume decreases, so if you have decreasing volume on the low end and perfectly flat phase it sounds really weird.

3) Other phase correction:

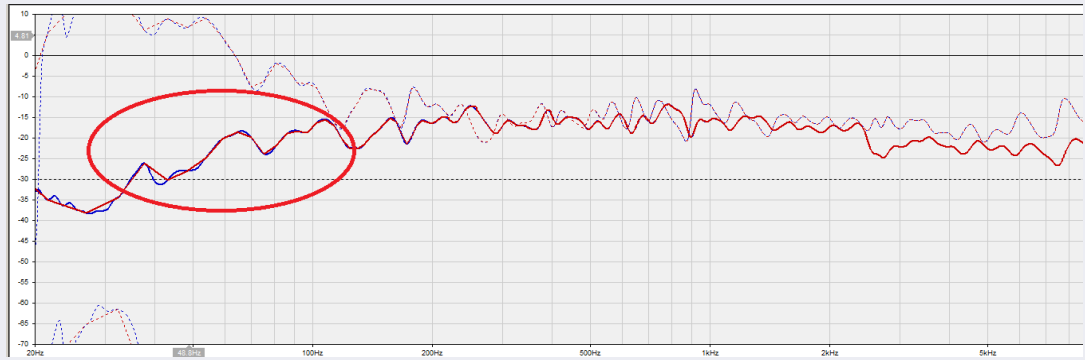
Next, I'd suggest looking through the graphs for any other places where there are changes in phase that don't correspond to changes in frequency, that is, any places where the phase appears to rise or fall, but the frequency response does not also rise or fall. If you identify any of these other phase anomalies (sometimes called "excess phase"), you might try flattening those out as well using the paragraphic phase EQ, but if they're small you might want to leave them alone. Phase is very location dependent in the room so you can wind up chasing phase phantoms all day if you overdo it. By contrast, if you have large phase non-linearities that **do** correspond with frequency response non-linearities, those are things that could potentially be addressed through additional regular EQ as described above (normal EQ is "minimum phase" and will adjust both frequency response and phase at the same time).

Making the filters:

The default settings in RePhase should be fine for the limited amount of phase correction we're talking about doing. If the phase response of your two speakers is similar enough that the same correction would work for either, be sure to select 32-bit LPCM stereo as your output format. If you want separate correction filters for each side, then make sure the output format is mono. If you made separate filters for both sides you'll need to write a config file for use in JRiver's convolution engine that routes your channels through the correct filters. An example config file for a 2-channel 44.1KHz filter would look like this:

```
44100 2 2 0
0 0
0 0
C:\[your filepath]\[your left channel filter filename].wav
0
0.0
0.0
C:\[your filepath]\[your right channel filter filename].wav
1
1.0
1.0
```

After you've generated the filter(s), make sure that the generated filter actually resembles the target response in RePhase.



You can see here that my filter deviates from the target response to some extent. If yours is imperfect, try increasing the number of taps and generating a new filter. If yours is perfect, you may want to try lowering the number of "taps" and generating another filter. The taps represent the length of the filter in time: the more taps there are, the more complex the filters can be, but also the more latency introduced by the resulting filter. If you can accomplish the filtering you need with a very small number of taps, you can reduce the latency introduced by the resulting filter, which will make it more useful to you (i.e. potentially usable with movies, etc.). If you need a larger number of taps, then you may want to set up zones so the convolution is only engaged when listening to audio (as the latency can make video or loopback applications unworkable).

Once you've gotten your filters generated the way you want them, you'd want to select your stereo WAV file (or your config file if you did different WAV's for each side) in JRiver's convolution module as shown:

DSP Studio (Living Room Audio)

- ☒ Output Format
- ☒ Volume Leveling
- ☐ Adaptive Volume
- ☐ Equalizer
- ☒ Convolution
- ☒ Parametric Equ...
- ☒ Parametric Equ...
- ☐ Effects
- ☐ Headphones
- ☐ Tempo & Pitch
- ☐ Room Correction Analyzer

Convolution

Playback stopped or current playback doesn't s... [Option](#)

Convolution enables sophisticated time and frequency adjustments.

Convolution requires a filter file tailored for your speakers and room. There are several third-party tools available to help make filter files.

Settings

Filter (impulse audio file or convolver text configuration):

☐ Normalize filter volume (recommended)

☐ Automatically switch filter based on input format (recommended)

Status

Processing: 2 paths
Performance: 0.0X real-time (using SSE)
Filter: C:\Users\Cato 6\Desktop\RePhase\Linearizationwithlumpflattened...

Processed in order listed (drag to reorder)

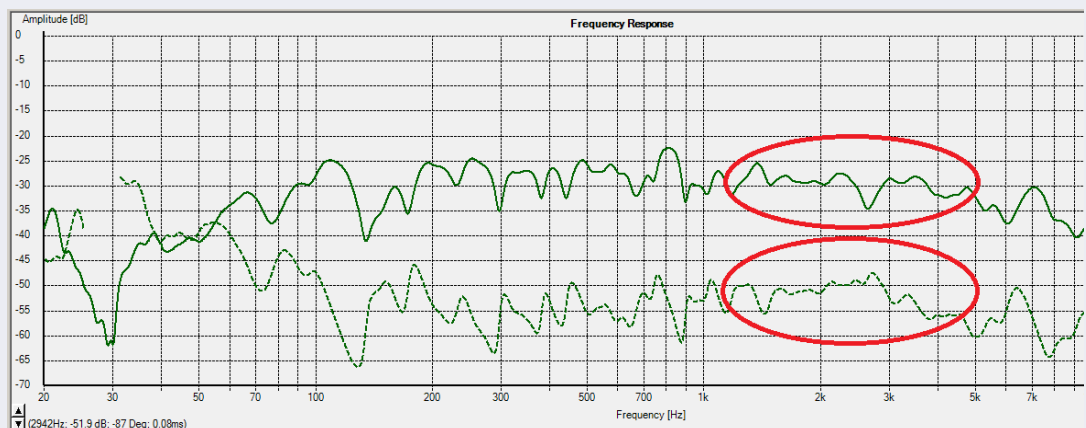
Clip protection

Peak Level: n/a

Source: n/a

Internal: n/a

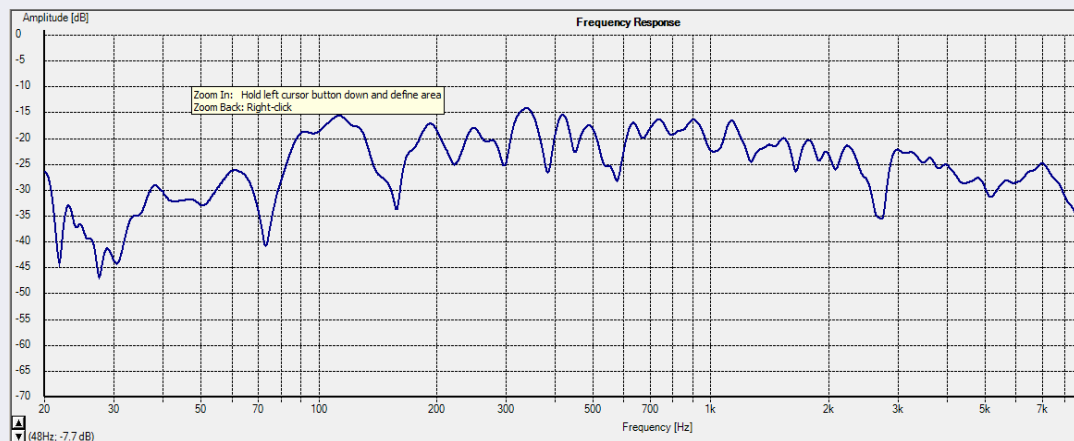
Lastly, take a measurement at ear/tweeter height from 1 meter with all of the filters engaged to make sure that it worked as expected.



Note that the phase in the crossover region is flattened as expected, and that the frequency null at the crossover is improved somewhat by ironing out the phase response there. Phase manipulation is the only way to even partially fix a goofed up passive crossover, but it can't resolve everything, obviously.

Results of speaker correction:

Now take a measurement at your listening position with the filters on and compare to your initial measurement. Here's mine:



It's still pretty wacky looking, especially around the crossover, but it's much better. And these results are almost entirely from correcting the speakers with very limited attempts to correct the room response. If you want to undertake additional correction targeted at the room's response, it's possible to make some additional gains, but there are a few pitfalls to avoid.

Logged

JRiver on Windows 10 64 bit || Arch Linux 64 bit with Gnome
Steinberg UR824 -> Homebrew Bi-Amped Mains ("Project Warhom")

Check out my Guides:

[Guide to Speaker Correction using JRiver](#)

[Guide to JRiver on the Raspberry Pi](#)

[Casting Audio to JRiver](#)

mwillems

MC Beta Team
Citizen of the Universe

Posts: 3580



"Linux Merit Badge"
Recipient

Re: Guide to Speaker/Room Correction Using Freeware and JRiver

« Reply #5 on: February 17, 2014, 12:12:48 pm »

Room correction:

If you want to go the next step of attempting to correct the room response (in addition to the speaker response), you can apply some of the same techniques covered above with a few modifications and caveats.

For example, you can (after doing your speaker correction), take measurements with frequency dependent windowing at each of the listening positions in the room, export them to REW, average them and then generate additional correction filters for the averaged response. However, you may find that the measurements at the various listening positions are different enough that even the average may not be that helpful. However they are likely to agree on at least some things, and those are things that may benefit from EQ. A few things to keep in mind:

1) Room nulls or boundary cancellation effects (large dips in the response caused by the room) can't usually be corrected by EQ. Attempted corrections are not likely to work, and will compromise the sound somewhere else in the room even if they do work. By contrast, peaks or room resonances (large raised lumps in the response) often can be usefully EQed out. If you want to account for this and "play it safe" in generating your correction, you can set the maximum boost in REW to 0dB, so that it only applies cuts and doesn't even attempt to boost. You can also limit boost to one or two dB as a less risk-averse compromise solution.

2) Because the wavelengths of high frequency sound are very short, attempts to EQ room behavior at high frequencies are a moving target. At very high frequencies, it's hard to even position the microphone consistently enough to get consistent measurements off axis. For this reason I'd recommend limiting attempts to EQ room behavior to frequencies below 1 KHz or so (this can be achieved by setting the top of the "match range" in REW to 1 KHz).

3) Some of the non-linearities you see in some listening positions may not even result from the room, but rather from the directivity of your loudspeakers. Loudspeakers are engineered to be as flat as possible on axis, but as you rotate off axis, the frequency response of the loudspeaker will begin to change, and will tend to change differently at different frequencies because the directivity of a loudspeaker tends to vary with frequency response. If your listening positions are significantly off-axis (i.e. more than 30 degrees off axis) from either speaker you will get nonlinearities that cannot meaningfully be corrected without messing things up for other listening positions because they are a consequence of the polar pattern of your loudspeaker. If all of your listening positions are significantly off-axis, you'd be better off rotating the speakers or otherwise repositioning them.

As you might've gathered, those caveats are pretty significant. The best way to test whether your proposed room correction is a net plus or minus is to re-measure at each listening position, and on axis with the tweeter at 1 meter. The listening positions should each look better (or at least not worse), and the on axis response shouldn't look completely crazy. If the correction fails those tests, you know you've done too much and need to scale it back.

Conclusions:

This obviously just begins to scratch the surface on a lot of issues, but this is a method that has produced repeatable positive results for me. There's always more to learn, and always more tweaking to be done, but hopefully this has been a useful beginning for some folks. Here are a few additional links that might be helpful to folks trying to feel their way through more advanced measurement and correction topics:

Megathread on Holm over at Diyaudio (where the software author posts once in a while):
<http://www.diyaudio.com/forums/software-tools/144984-holmimpulse-measuring-frequency-impulse-response.html>

Very detailed REW Manual: http://www.hometheatershack.com/roomeq/REWV5_help.pdf (the hometheatershack forums are also a great resource on REW)

Mojave's guide to using REW as a real-time analyzer with JRiver (very useful when you're tweaking your settings by hand): <http://yabb.jriver.com/interact/index.php?topic=69725.0>

RePhase megathread over at DIYAudio (where the software author posts frequently):
<http://www.diyaudio.com/forums/multi-way/221434-rephase-loudspeaker-phase-linearization-eq-fir-filtering-tool.html>

Feel free to offer corrections, suggested improvements, or questions below ☐

☐ Logged

JRiver on Windows 10 64 bit || Arch Linux 64 bit with Gnome
Steinberg UR824 -> Homebrew Bi-Amped Mains ("Project Warhorn")

Check out my Guides:
[Guide to Speaker Correction using JRiver](#)
[Guide to JRiver on the Raspberry Pi](#)
[Casting Audio to JRiver](#)

JimH

Administrator
Citizen of the Universe



Posts: 46907



Miller drives a tall-masted tractor on the ocean

☐ **Re: Guide to Speaker/Room Correction Using Freeware and JRiver**

« **Reply #6 on:** February 17, 2014, 12:32:16 pm »

mwillems,
Thanks very much for your work on this.

Jim

☐ Logged

JRiver, Inc
[Getting Started](#) [Up and Running](#) [FAQ](#) [Weird problems](#) [License Info](#) [Wiki](#) [Sound Problems](#)

mojave

MC Beta Team
Citizen of the Universe



Posts: 3359

Requires "iTunes or better" so I installed JRiver

☐ **Re: Guide to Speaker/Room Correction Using Freeware and JRiver**

« **Reply #7 on:** February 17, 2014, 01:58:29 pm »

Thanks for writing this up. ☐ I have a few questions, but want to reread your posts and think about them some more.

I noticed you mentioned using MME output in HOLM. MME is for legacy software prior to Windows XP. MME only supports 16 bit and 44.1 KHz output. Although HOLM shows other sample rates, the [Wikipedia article](#) says MME resamples everything to 44.1. I would recommend at least DirectSound. DirectSound will still use the loopback and has lower latency.

Where did you find examples for how to do a convolution config file? I've been wanting to do this for a while, but never got around to it. REW lets you export the filters' impulse response as a wav. If not using Rephase, one could export their REW filters and create a config file rather than manually entering in all the filters into JRiver. Since a lot of people just EQ the subwoofer channel, the config file needs either only reference channel 4 or show the other channels as blank.

☐ Logged

mwillems

MC Beta Team
Citizen of the Universe



Posts: 3580



"Linux Merit Badge" Recipient

☐ **Re: Guide to Speaker/Room Correction Using Freeware and JRiver**

« **Reply #8 on:** February 17, 2014, 02:11:38 pm »

Quote from: mojave on February 17, 2014, 01:58:29 pm

Thanks for writing this up. ☐ I have a few questions, but want to reread your posts and think about them some more.

I noticed you mentioned using MME output in HOLM. MME is for legacy software prior to Windows XP. MME only supports 16 bit and 44.1 KHz output. Although HOLM shows other sample rates, the [Wikipedia article](#) says MME resamples everything to 44.1. I would recommend at least DirectSound. DirectSound will still use the loopback and has lower latency.

That's a very good point, I had been using MME, but direct sound is more flexible. I'll edit it to recommend direct sound.

Quote

Where did you find examples for how to do a convolution config file? I've been wanting to do this for a while, but never got around to it. REW lets you export the filters' impulse response as a wav. If not using Rephase, one could export their REW filters and create a config file rather than manually entering in all the filters into JRiver. Since a lot of people just EQ the subwoofer channel, the config file needs either only reference channel 4 or show the other channels as blank.

That's adapted from the config file that I personally use for two-channel convolution on my office stereo (which is the one I was using as the example in the guide). This guide is pitched towards getting the most out of a pair of bookshelf speakers, so I figured the config file should match that goal. I'm planning to do a later guide to deal with subs and active crossovers specifically, and your point about the subs is a good one. I clarified that the config file is only for a 2-channel setup, and I'll make sure to provide less generic configs in the later guide.

I do briefly mention above that REW allows for WAV export, but I don't spend much time on it, because I don't personally recommend it (even if one isn't using RePhase). By using PEQ for the correction (instead of convolution), one can use the bulk of one's speaker correction with loopback, DVD sources, video games, etc. with no trouble. That's not possible if you export the filters as a WAV and use them in the convolution

module. So from my perspective, the preferred approach is to copy the filters into PEQ so that folks can get the most use out of the correction they've generated. I'll beef up the export discussion a little, though, so the option is clearer.

Thanks very much for the feedback; keep it coming ☐

☐ Logged

JRiver on Windows 10 64 bit || Arch Linux 64 bit with Gnome
Steinberg UR824 -> Homebrew Bi-Amped Mains ("Project Warhorn")

Check out my Guides:
[Guide to Speaker Correction using JRiver](#)
[Guide to JRiver on the Raspberry Pi](#)
[Casting Audio to JRiver](#)

mojave

MC Beta Team
Citizen of the Universe



Posts: 3359

☐

Requires "iTunes or better" so I installed JRiver

☐ **Re: Guide to Speaker/Room Correction Using Freeware and JRiver**

« **Reply #9 on:** February 17, 2014, 02:45:04 pm »

Quote from: mwillems on February 17, 2014, 02:11:38 pm

That's adapted from the config file that I personally use for two-channel convolution on my office stereo (which is the one I was using as the example in the guide). This guide is pitched towards getting the most out of a pair of bookshelf speakers, so I figured the config file should match that goal.

I'm really asking, "Where on the internet does one find the proper format for a convolution configuration file?"

Edit: I think I found it here: [Config file](#)

☐ Logged

mattkhan

MC Beta Team
Citizen of the Universe



Posts: 762

☐ **Re: Guide to Speaker/Room Correction Using Freeware and JRiver**

« **Reply #10 on:** February 17, 2014, 02:47:54 pm »

Quote from: mojave on February 17, 2014, 01:58:29 pm

Where did you find examples for how to do a convolution config file?

Since a lot of people just EQ the subwoofer channel, the config file needs either only reference channel 4 or show the other channels as blank.

The reference page describes the format and there is a linked examples page that gives you a start on how to use it.

<http://convolver.sourceforge.net/config.html>

This mentions that unfiltered output channels get fed with silence by convolver. Does jriver work differently?

IIRC jriver uses this format but does not support certain features, this post has details

<http://yabb.jriver.com/interact/index.php?topic=68828.msg465919#msg465919>

Quote from: mwillems on February 17, 2014, 12:03:41 pm

Now you'll take a series of measurements, as follows:

I recommend making five measurements for a two-way speaker: 1) on axis with the woofer from three inches away, 2) on axis with the tweeter from three inches away, 3) on axis with the woofer from one foot away, 4) on axis with the tweeter from one foot away, and finally 5) on axis with the tweeter from one meter away.

You should do the same "stitching" process to the one foot measurements as well. You should now have three "final" measurements for your speaker: a composite three-inch measurement, a composite one-foot measurement, and a single one-meter measurement.

The three inch measurement shows the direct output of the driver with very little else (which can be deceptive, because the effect of the edges of the speaker cabinet itself are not always captured with a measurement that close). The one foot measurement shows the output of the driver along with some baffle diffraction (sound reflecting off the sides of the speaker cabinet) and a little bit of the room. The one meter measurement shows more of the room's effect, as you can no doubt see, as well as the interaction between the two drivers at ear height.

Once you have exported all the measurements, open up Room EQ Wizard. REW has a measurement suite as well, but it does not support frequency dependent windowing or certain other features that Holm supports, so we will not be doing any measurements in REW in this guide. Instead, select "import frequency response" from the file menu and import the three measurements for your first speaker in REW. Tell it not to do a C-weighting correction, unless your mic is C-weighted (most calibrated mics are not).

Once you have the three measurements for your first speaker imported, click the "All SPL" button along the top. Make sure all three are displayed, and then click on the "average the responses" button.

Thanks for posting this, a couple of questions if I may. I snipped a few relevant bits from a post to avoid a mega quote.

1) why do you recommend averaging the response from different distances? If the aim is to correct/linearise the speaker then why not just correct the (very) near field measurement? I think I find the inclusion of the 1m measurement particularly confusing in this regard

2) do you have any views on how to take these measurements if you have a >1 driver in each way?

☐ Logged

mwillems

MC Beta Team
Citizen of the Universe



Posts: 3580



"Linux Merit Badge" Recipient

☐ **Re: Guide to Speaker/Room Correction Using Freeware and JRiver**

« **Reply #11 on:** February 17, 2014, 03:11:37 pm »

Quote

1) why do you recommend averaging the response from different distances? If the aim is to correct/linearise the speaker then why not just correct the (very) near field measurement? I think I find the inclusion of the 1m measurement particularly confusing in this regard

It's a compromise. The very close measurement shows the *driver's* response, but in a sense it doesn't completely show the *speaker's* response (i.e. the response of the driver in that specific baffle). The one foot measurements are necessary because very close measurements effectively gate out baffle diffraction and room gain that might offset baffle-step type losses. I recognize that the one meter measurement has some potential problems, but I recommend including it because, empirically, I've gotten better results when I included it. I theorize that to be the case for four reasons:

- 1) If your speaker's baffle is large (i.e. larger than one foot on a side), the 1 foot measurement may not fully capture the baffle diffraction effects.
- 2) Depending on driver size or spacing, closer measurements may not fully show how the two drivers interact at the crossover
- 3) Rear firing ports (which are quite common) often don't measure correctly when closer measurements are taken, and
- 4) Baffle step compensation needs to be taken into account when correcting a speaker's response, but the amount of compensation needed will be dependent on the speaker's placement in the room (i.e. room gain from corner placement, or vice versa). This is more clearly visible at 1 meter.

Taking a third measurement farther away also minimizes the chance that the other two measurements are affected by a localized room effect that isn't present farther out in the room. But, to some extent, those are just my best guesses based on some differences I've seen in measurements at different distances. I do know that my final correction tends to measure better when I include the 1 meter measurement in the average (and sounds better too).

The disadvantage of measuring at 1 meter, obviously, is that you get some of the room in the measurement. I think the "damage" is limited by the fact that frequency dependent windowing gates out some of the worst mischief, and that the measurement is averaged with two other more "reliable" measurements. It may well be that I'm, in effect, doing some limited room correction by including the 1 meter measurement (room gain is certainly a type of room effect).

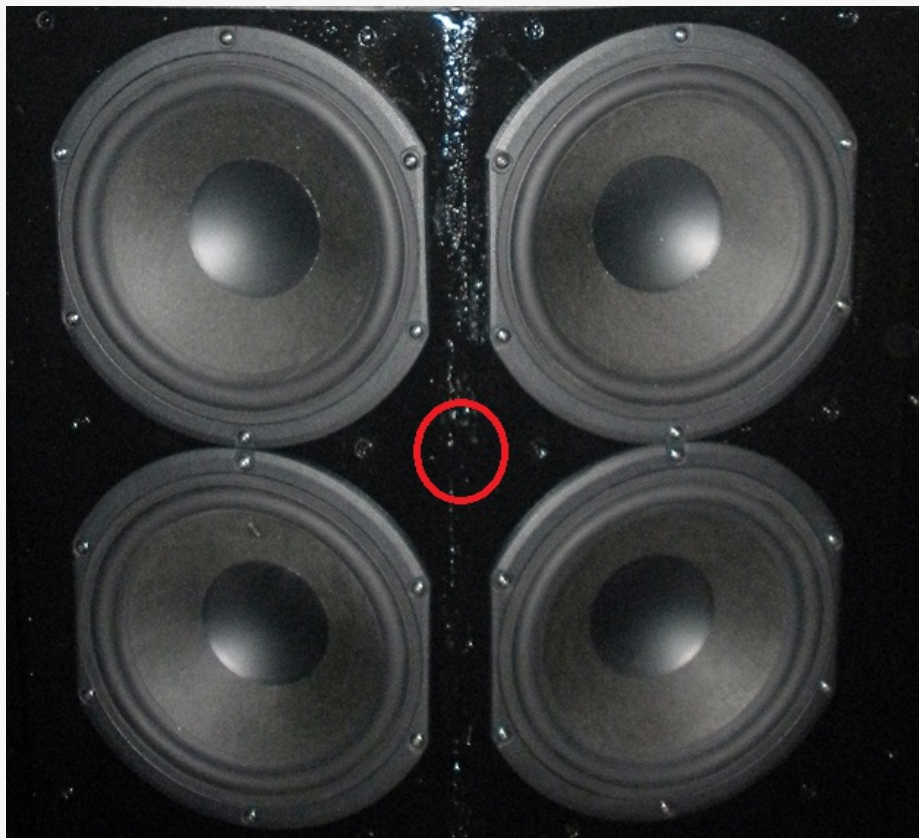
Try it both ways and see where you land. I would guess that if you are measuring a fairly small (<1 cubic foot) sealed speaker, you could probably safely leave out the 1 meter measurement and still get good results.

[above edited for clarity]

Quote

2) do you have any views on how to take these measurements if you have a >1 driver in each way?

I went through this myself with my larger bi-amped speakers. You want to target the "acoustic center" of the stage in question. So for the mid-basses in the big speakers, which are a quad array, I aligned the microphone to the point on the baffle that marked the geometric "center" of the four elements:



Other driver arrangements might be tougher. For example, I haven't personally measured an MTM, but I bet the 3 inch measurement would be tricky. If I were measuring one, I'd start by trying to measure from the acoustic center (i.e. half way between the woofers). If that didn't work, I'd just take the three inch measurement from the acoustic center of one of the two elements. It might be the case for an MTM that you don't even need to go through the stitching process because the acoustic center of the low frequency stage is often co-located with the acoustic center of the high frequency stage.

Quote from: [mojave](#) on February 17, 2014, 02:45:04 pm

I'm really asking, "Where on the internet does one find the proper format for a convolution configuration file?"

Edit: I think I found it here: [Config file](#)

That's what I used to write my config file

▢ Logged

Check out my Guides:
[Guide to Speaker Correction using JRiver](#)
[Guide to JRiver on the Raspberry Pi](#)
[Casting Audio to JRiver](#)

jdubs

Junior Woodchuck
Posts: 81

Re: Guide to Speaker/Room Correction Using Freeware and JRiver
« Reply #12 on: February 17, 2014, 08:26:25 pm »

SWEET!!! Thank you so much for this!!

-Jim

Logged

Mitchco

MC Beta Team
Junior Woodchuck
Posts: 97

Re: Guide to Speaker/Room Correction Using Freeware and JRiver
« Reply #13 on: February 18, 2014, 12:47:24 am »

A well written guide mwillems! Cheers!

Mitch

Logged

newguy1

Recent member
Posts: 44

Re: Guide to Speaker/Room Correction Using Freeware and JRiver
« Reply #14 on: March 22, 2014, 10:45:11 pm »

Firstly, thanks for taking the time to write this!

I have a ported there way, so I'm guessing that I should measure the tweeter, mid, woofer and port at 3", 1' and listening position, then stitch them together?

Thanks for your help!

David

Logged

mwillems

MC Beta Team
Citizen of the Universe
Posts: 3580



"Linux Merit Badge"
Recipient

Re: Guide to Speaker/Room Correction Using Freeware and JRiver
« Reply #15 on: March 23, 2014, 07:54:47 am »

Quote from: newguy1 on March 22, 2014, 10:45:11 pm

Firstly, thanks for taking the time to write this!

I have a ported there way, so I'm guessing that I should measure the tweeter, mid, woofer and port at 3", 1' and listening position, then stitch them together?

Thanks for your help!

David

Depending on the positioning and tuning of your port, you may not need to measure the port separately at all. If you do need to measure it separately, I'd recommend doing separate measurements only at 3 inches or 1 foot, but not doing a separate measurement for the port at one meter or the listening positions.

If your ports are front-firing, tuned low, and near the woofers, you may not need to measure them separately; low frequency wavelengths can be enormous. For example, I have some cabinets with front firing ports tuned to 28 Hz that are just below the woofers. I didn't need to measure those ports separately because they're tuned low enough and are close enough to the woofers that the port output was "showing up" in the woofer measurements even at 3 inches. The wavelength at 28Hz is about 40 feet, so the six or eight inches between my drivers and my ports wasn't really relevant.

If you have, instead, a rear-firing port tuned much higher, you may well need to measure it separately for the three inch measurement and maybe the 1 foot measurement. See if it "shows up" in the 1 foot woofer measurement before you measure it separately. If you're not sure what I mean about the port "showing up," I just mean that when you take the 3 inch and 12 inch measurements of the woofer, does the woofer measure as low and as loud as the speaker is rated to go? If so you're getting the port output in your measurements.

If it doesn't get quite as low or as loud, you're only getting partial pickup and you might want to measure the port separately. Another test to see if you need to measure separately at 1 foot: do you get noticeably better bass extension on the 1 meter measurement than the 1 foot measurement? If so, that suggests that you might not be getting all of your port output in the 1 foot close measurement (although that's not ironclad, there could be other reasons).

If you do wind up needing to measure the port separately, just stitch it to the other measurements in the same way I recommended stitching the woofer and the tweeter above. It should be relatively obvious where the best stitch frequency is, but a good place to start looking is about 1 octave above the port tuning.

I recommend only taking a single measurement at 1 meter or the listening position, because a measurement at those distances should catch basically all speaker and port output on most home speakers (a hypothetical 10 or 20-foot tall speaker might need special treatment, etc.). There's usually no reason to measure the port separately that far out because the speaker is designed to present integrated sound in the mid to far field.

You'll note that I don't even recommend measuring the woofer and the tweeter separately at 1 meter or the listening position; that's because I think it's important to take at least one measurement that shows how the elements actually interact with each other at the crossovers (or the changeover from direct to reflex sound in the case of the port), rather than relying solely on recreating the hypothetical interaction by stitching.

Logged

JRiver on Windows 10 64 bit || Arch Linux 64 bit with Gnome
Steinberg UR824 -> Homebrew Bi-Amped Mains ("Project Warhorn")

Check out my Guides:
[Guide to Speaker Correction using JRiver](#)
[Guide to JRiver on the Raspberry Pi](#)
[Casting Audio to JRiver](#)

shawns1234

Member

Posts: 4

Re: Guide to Speaker/Room Correction Using Freeware and JRiver

« Reply #16 on: April 21, 2014, 01:13:24 pm »

Let me ask a potentially stupid question. I currently bitstream my audio to Marantz 8002 or 8200...cant remember atm.. in order to use this technique, do I have to let JRiver decode my bluray audio to implement these adjustments? I would assume so. If so, can I get a consensus on if the audio will sound as good? Ive never had good results with letting computer software decode hi def tracks and presenting as multichannel as I have letting my receiver do the work.

I do have a stand alone parametric eq for my subs.

Thanks

Any other wisdom on this matter is welcomed.

Logged

mwillems

MC Beta Team
Citizen of the Universe

Posts: 3580



"Linux Merit Badge"
Recipient

Re: Guide to Speaker/Room Correction Using Freeware and JRiver

« Reply #17 on: April 21, 2014, 05:19:45 pm »

Quote from: shawns1234 on April 21, 2014, 01:13:24 pm

Let me ask a potentially stupid question. I currently bitstream my audio to Marantz 8002 or 8200...cant remember atm.. in order to use this technique, do I have to let JRiver decode my bluray audio to implement these adjustments? I would assume so. If so, can I get a consensus on if the audio will sound as good? Ive never had good results with letting computer software decode hi def tracks and presenting as multichannel as I have letting my receiver do the work.

I do have a stand alone parametric eq for my subs.

Thanks

Any other wisdom on this matter is welcomed.

Almost all JRiver DSP is disabled when you're bitstreaming, so the short answer is that you'd need to disable bitstreaming and do the decoding in JRiver to use this guide. I've always done all my decoding in JRiver (I don't have an AVR), so I don't have a frame of comparison, but my experience has been very, very good. My sense from talking to other folks here on the forum who've tried it both ways is that JRiver's decoding is top notch.

If you've had bad luck getting JRiver's decoding specifically configured to your taste in the past, I'd suggest opening a topic about it on the forum. The community is very active and there are lots of folks here who know much more about encoding and decoding than I do (including the developers).

Logged

JRiver on Windows 10 64 bit || Arch Linux 64 bit with Gnome
Steinberg UR824 -> Homebrew Bi-Amped Mains ("Project Warhorn")

Check out my Guides:
[Guide to Speaker Correction using JRiver](#)
[Guide to JRiver on the Raspberry Pi](#)
[Casting Audio to JRiver](#)

bluescale

Recent member

Posts: 19

Re: Guide to Speaker/Room Correction Using Freeware and JRiver

« Reply #18 on: May 04, 2014, 02:47:27 pm »

First of all, let me just say that this guide is so awesome, that it finally convinced me to come out of lurker mode and make my first post to this forum. Thank you so much for puttin this together. It's a lot of great information for a newb like me to absorb, and it's presented in a way I can actually absorb it.

Now my question. I'm still unclear why it's necessary to use HOLMImpulse rather than REW for this process. I'm trying to do this for my home theater, and I get the impression HOLM is limited to L/R channels.

Also, you recommend using Direct mode in HOLM. However, doesn't direct mode pass all sounds through Window's processing? Is there a reason not to use ASIO? I'm assuming (and perhaps I should just download the software and find out for sure), that you could have it use JRiver as the ASIO driver and just pass sounds through that way - similar to what I can do with REW.

Logged

mwillems

MC Beta Team
Citizen of the Universe

Posts: 3580



"Linux Merit Badge"
Recipient

Re: Guide to Speaker/Room Correction Using Freeware and JRiver

« Reply #19 on: May 04, 2014, 03:22:57 pm »

Quote from: bluescale on May 04, 2014, 02:47:27 pm

First of all, let me just say that this guide is so awesome, that it finally convinced me to come out of lurker mode and make my first post to this forum. Thank you so much for puttin this together. It's a lot of great information for a newb like me to absorb, and it's presented in a way I can actually absorb it.

Thanks for the kind words!

Quote

Now my question. I'm still unclear why it's necessary to use HOLMImpulse rather than REW for this process. I'm trying to do this for my home theater, and I get the impression HOLM is limited to L/R channels.

The main reason I don't recommend using REW for measurement in this guide is that it doesn't offer frequency dependent windowing (FRD), which is very important to how the guide works. FRD filtering filters out a lot of room echoes and boundary effects and produces much, much more consistent measurements (i.e. it's much easier to get repeatable and comparable measurements with FRD, especially when you're moving your mic around). In addition to added measurement accuracy and consistency, there's some evidence that our ears process sound in a way that's analogous to FRD (i.e. our ears/brains ignore certain "late echoes"), so FRD also produces a measurement that looks a little more like what we hear than a conventional "Raw" measurement.

I've also been unimpressed with REW's measurement accuracy overall, especially in the low frequency range. I often get low frequency measurements in REW that don't agree with measurements taken from the same position from several other measurement suites. Check out this recent thread for someone else running into the same problem: <http://yabb.jriver.com/interact/index.php?topic=88942.msg610271#msg610271>

Holm also offers some additional handy features not immediately relevant to this guide, like the ability to time-lock a measurement and measure the time offset between it and another measurement, which is great for setting your delays in a multichannel system (out of the scope of this guide, but still neat).

As for the channel output, Holm can natively run sweeps on up to 4 channels, but it's very easy to use JRiver's Parametric EQ to temporarily re-route the sweeps to any channel you want by using the "copy" or "move" functions under "channel mixing." If you run into trouble with that, let me know and I can provide a step by step.

Quote

Also, you recommend using Direct mode in HOLM. However, doesn't direct mode pass all sounds through Window's processing? Is there a reason not to use ASIO? I'm assuming (and perhaps I should just download the software and find out for sure), that you could have it use JRiver as the ASIO driver and just pass sounds through that way - similar to what I can do with REW.

The only reason I didn't recommend ASIO is because it's a little more complicated to setup and not everyone has an ASIO compatible soundcard for the input (i.e. folks with a USB mic). If you're comfortable with ASIO setup there's absolutely no reason not to use ASIO, and it definitely has some advantages (I've used it myself).

▢ Logged

JRiver on Windows 10 64 bit || Arch Linux 64 bit with Gnome
Steinberg UR824 -> Homebrew Bi-Amped Mains ("Project Warhorn")

Check out my Guides:
[Guide to Speaker Correction using JRiver](#)
[Guide to JRiver on the Raspberry Pi](#)
[Casting Audio to JRiver](#)

bluescale

Recent member

Posts: 19

Re: Guide to Speaker/Room Correction Using Freeware and JRiver

« Reply #20 on: May 17, 2014, 03:58:20 pm »

Unfortunately, I'm having a really hard time getting JRiver to behave in such a way that I can even get started with measurements. When I try to use ASIO as my driver, JRiver mixes the audio to 2 channel. In other words, I can't get it play the signal through a single channel at a time. When I look in Audio Paths, it clearly shows that it's mixing from 1 channel to 2. However, when I go into the DSP studio, it's set to source number of channels. If I try to use the loopback functionality, I get a really weird, choppy sound that is somewhat like reverb.

Any thoughts on what I could be doing wrong?

Edit - I should also point out that I tried bitstreaming to see if that would output just 1 channel of sound at a time, but alas, no luck.

▢ Logged

mwillems

MC Beta Team
Citizen of the Universe

Posts: 3580



"Linux Merit Badge"
Recipient

Re: Guide to Speaker/Room Correction Using Freeware and JRiver

« Reply #21 on: May 17, 2014, 07:09:31 pm »

Quote from: bluescale on May 17, 2014, 03:58:20 pm

Unfortunately, I'm having a really hard time getting JRiver to behave in such a way that I can even get started with measurements. When I try to use ASIO as my driver, JRiver mixes the audio to 2 channel. In other words, I can't get it play the signal through a single channel at a time. When I look in Audio Paths, it clearly shows that it's mixing from 1 channel to 2. However, when I go into the DSP studio, it's set to source number of channels. If I try to use the loopback functionality, I get a really weird, choppy sound that is somewhat like reverb.

My first suggestion would be to try not using ASIO as the output (i.e. using direct sound)? It usually works "out of the box" and might help identify what the issue is if that works.

I haven't had the problems you're describing with ASIO, but did you try just muting one of the channels in DSP studio? If all it's doing is cloning the channel, that should work fine. You can do almost infinitely flexible channel routing in JRiver.

I've never had "reverb" with loopback, but make sure that you have loopback setup correctly (i.e. that your default windows output device is not the same output device that JRiver is trying to output to). If you try to loopback from the same device JRiver is trying to use as an output, you'll get echo for sure (or it will throw errors). If that's not the problem, try changing the loopback buffer sizes. Here's a step by step on setting up loopback:

1. To use the loopback you have to have another soundcard in the system. You can use the motherboard soundcard. Nothing is actually connected to it, but the drivers still need to be installed. You might need to turn autosense off in the motherboard drivers. Many people have the Realtek HD Audio on their motherboard. Disabling front jack detection in the Realtek HD Audio Manager is what actually enables the analog output from front and rear outputs.
2. Set the above soundcard as 7.1 (with fullrange speakers) and as the default soundcard in the Windows Control Panel.
3. Set JRiver's Audio output [to whatever your soundcard is]
4. Before starting Holm, put JRiver in loopback mode by going to File > Open URL and enter live://loopback [or open live, WASAPI Loopback]

(adapted from an older post by mojave to make it applicable to the current JRiver)

Quote

Any thoughts on what I could be doing wrong?

Edit - I should also point out that I tried bitstreaming to see if that would output just 1 channel of sound at a time, but alas, no luck.

I would strongly suggest (as I alluded to above) either using direct sound or just using JRiver's DSP studio to handle channel routing. Direct sound isn't ideal for normal audio output, but it doesn't seem to affect measurements (or at least these kinds of measurements) much. I can get identical measurements using either output mode.

Barring that, you can use a combination of muting and channel copying to put the sweeps wherever they need to be.

▢ Logged

JRiver on Windows 10 64 bit || Arch Linux 64 bit with Gnome
Steinberg UR824 -> Homebrew Bi-Amped Mains ("Project Warhorn")

Check out my Guides:
[Guide to Speaker Correction using JRiver](#)
[Guide to JRiver on the Raspberry Pi](#)
[Casting Audio to JRiver](#)

bluescale

Recent member

Posts: 19

Re: Guide to Speaker/Room Correction Using Freeware and JRiver

« Reply #22 on: May 18, 2014, 12:59:40 am »

Quote from: mwillems on May 17, 2014, 07:09:31 pm

I would strongly suggest (as I alluded to above) either using direct sound or just using JRiver's DSP studio to handle channel routing. Direct sound isn't ideal for normal audio output, but it doesn't seem to affect measurements (or at least these kinds of measurements) much. I can get identical measurements using either output mode.

Barring that, you can use a combination of muting and channel copying to put the sweeps wherever they need to be.

Let me say again, just how much I appreciate your help. It's been invaluable, and I've learned (and am learning) a quite a lot. I'm still not sure why using JRiver's ASIO driver as the output from Holm didn't work, but I never got that function successfully. In the end, the time I spent trying to get it to work was a fools errand anyway. I went back and reread what you mentioned to me previously, and since I'm running a USB mic, it was a no go.

In case anyone else runs into this issue, I'll put forth what eventually worked for me. It's basically in line with what you suggested, but I think I just had to bumble through it on my own to figure out how things worked:

1. Windows default configured to output audio via the onboard HDMI port
2. JRiver configured to output using ASIO4ALL
3. ASIO4ALL configured to output to my Nvidia video card
4. Loopback configured using Open Live, WASAPI Loopback (I could never get it to work using Open URL)
5. DSP Studio configured to 5.1, but no upmixing or downmixing
6. JRiver PEQ configured to map sound to the appropriate channels
7. Holm configured to use the onboard HDMI port (i.e., same as the Windows default)

If in step 2 if I tried to configure JRiver to use WASAPI and output directly to my Nvidia card, I'd get the choppy, reverb sound every time.

As a test, I've gently EQ'd my sub using a simplified version of your steps. The results were quite a bit better than what the auto EQ in my processor produced. Now I'm going to start playing with the L/R/C channel speakers.

Thanks for your time and patience. I'm finally getting to the fun part .

▢ Logged

mwillems

MC Beta Team
Citizen of the Universe

Posts: 3580



"Linux Merit Badge"
Recipient

Re: Guide to Speaker/Room Correction Using Freeware and JRiver

« Reply #23 on: May 18, 2014, 08:30:33 am »

Quote from: bluescale on May 18, 2014, 12:59:40 am

As a test, I've gently EQ'd my sub using a simplified version of your steps. The results were quite a bit better than what the auto EQ in my processor produced. Now I'm going to start playing with the L/R/C channel speakers.

Thanks for your time and patience. I'm finally getting to the fun part .

I'm happy to help and I'm glad you got it working!

The ASIO4ALL piece is odd; direct WASAPI output in JRiver shouldn't produce strange choppy or echoing sound. If I had to guess it might be a buffer issue (whether the loopback buffer, or the normal buffer settings). If you have any interest in troubleshooting that WASAPI output further, you should open up a thread over on the main board. Lots of folks on the forums will probably have additional good ideas, and we can probably get to the bottom of it.

Also, if you need any advice about setting your sub to mains crossover and delay, just holler. I've been working (slowly) on a guide to bi-amping and sub integration; it's not ready yet but I can talk to specific "pain points" if you have any trouble.

▢ Logged

JRiver on Windows 10 64 bit || Arch Linux 64 bit with Gnome
Steinberg UR824 -> Homebrew Bi-Amped Mains ("Project Warhorn")

Check out my Guides:
[Guide to Speaker Correction using JRiver](#)
[Guide to JRiver on the Raspberry Pi](#)
[Casting Audio to JRiver](#)

bluescale

Recent member

Posts: 19

Re: Guide to Speaker/Room Correction Using Freeware and JRiver

« Reply #24 on: May 18, 2014, 12:23:48 pm »

Quote from: mwillems on May 18, 2014, 08:30:33 am

I'm happy to help and I'm glad you got it working!

The ASIO4All piece is odd; direct WASAPI output in JRiver shouldn't produce strange choppy or echoing sound. If I had to guess it might be a buffer issue (whether the loopback buffer, or the normal buffer settings). If you have any interest in troubleshooting that WASAPI output further, you should open up a thread over on the main board. Lots of folks on the forums will probably have additional good ideas, and we can probably get to the bottom of it.

At this point, I think I'm happy to just use it now that it's working.

Quote from: mwillems on May 18, 2014, 08:30:33 am

Also, if you need any advice about setting your sub to mains crossover and delay, just holler. I've been working (slowly) on a guide to bi-amping and sub integration; it's not ready yet but I can talk to specific "pain points" if you have any trouble.

As I mentioned in the other thread, I think I've got some issues going on at the crossover point. How do you feel about me posting a few graphs to get some input from you? Would you rather I do that in another thread?

Logged

mwillems

MC Beta Team
Citizen of the Universe
Posts: 3580



"Linux Merit Badge"
Recipient

Re: Guide to Speaker/Room Correction Using Freeware and JRiver

« Reply #25 on: May 18, 2014, 01:16:07 pm »

Quote from: bluescale on May 18, 2014, 12:23:48 pm

At this point, I think I'm happy to just use it now that it's working.

As I mentioned in the other thread, I think I've got some issues going on at the crossover point. How do you feel about me posting a few graphs to get some input from you? Would you rather I do that in another thread?

I replied in the other thread; I'm happy to look at a couple graphs, but it's probably better to do it in that thread rather than this one if that's ok.

Logged

JRiver on Windows 10 64 bit || Arch Linux 64 bit with Gnome
Steinberg UR824 -> Homebrew Bi-Amped Mains ("Project Warhorn")

Check out my Guides:
[Guide to Speaker Correction using JRiver](#)
[Guide to JRiver on the Raspberry Pi](#)
[Casting Audio to JRiver](#)

bluescale

Recent member
Posts: 19

Re: Guide to Speaker/Room Correction Using Freeware and JRiver

« Reply #26 on: May 19, 2014, 01:15:48 pm »

I need to play with things a bit more before I start posting graphs (in the other thread). I'll just say that things got WAY better between 120hz and 70hz when I disabled everything in my processor, and started using JRiver.

I have a question about measuring output from a sub. Since low frequency sound waves are considerably less directional, is it still necessary to go through the multi-distance setup you recommend in this thread? My sub is down-firing. With that in mind, if you still suggest using the procedure listed in this thread, how would you recommend I go about accomplishing that? I'd really rather not flip it on its side if at all avoidable, since the room has a concrete floor, and the sub's finish is piano gloss. It's not an insurmountable problem, but an annoyance.

Logged

mwillems

MC Beta Team
Citizen of the Universe
Posts: 3580



"Linux Merit Badge"
Recipient

Re: Guide to Speaker/Room Correction Using Freeware and JRiver

« Reply #27 on: May 19, 2014, 02:42:02 pm »

Quote from: bluescale on May 19, 2014, 01:15:48 pm

I need to play with things a bit more before I start posting graphs (in the other thread). I'll just say that things got WAY better between 120hz and 70hz when I disabled everything in my processor, and started using JRiver.

I have a question about measuring output from a sub. Since low frequency sound waves are considerably less directional, is it still necessary to go through the multi-distance setup you recommend in this thread? My sub is down-firing. With that in mind, if you still suggest using the procedure listed in this thread, how would you recommend I go about accomplishing that? I'd really rather not flip it on its side if at all avoidable, since the room has a concrete floor, and the sub's finish is piano gloss. It's not an insurmountable problem, but an annoyance.

It's not the directionality as much as the room effects that the methodology is trying to deal with. I think for best results you'd want to at least get one close measurement with the sub not pointed at the ground. If your sub is a large driver (12 or 15 inch) and is operating up at 120Hz, it's probably pretty directional up there. If it's only operating below 60 Hz, it probably won't matter as much. I also have a piano gloss sub, I usually put a thick towel on the floor to set it on when I flip it over.

If flipping it on its side even for the close measurement isn't an option, you should do the 3 inch measurement by positioning the mic directly under the sub (between the driver and the floor). Ideally you'd want to put it at 90 degrees and use a 90 degree calibration (as most mics tend to need more calibration at the extremes).

For the other more distant measurements, I'd suggest measuring it as it lies (downward firing), but just move 1 foot and 1 meter away in the direction of the listening position. You want to get those measurements with the sub in its normal position. If you see really drastic variation between the measurements, let me know and I'll offer some alternative suggestions.

Logged

JRiver on Windows 10 64 bit || Arch Linux 64 bit with Gnome
Steinberg UR824 -> Homebrew Bi-Amped Mains ("Project Warhorn")

Check out my Guides:
[Guide to Speaker Correction using JRiver](#)
[Guide to JRiver on the Raspberry Pi](#)
[Casting Audio to JRiver](#)

bluescale

Recent member

Posts: 19

Re: Guide to Speaker/Room Correction Using Freeware and JRiver

« Reply #28 on: May 19, 2014, 03:17:22 pm »

Final question (for now) - Since the goal is to get an idea of the speaker's frequency response, is it fair to assume that I should disable all bass management and other room correction before taking the measurements? I've been operating under that assumption, but it's not explicitly stated in the guide above. Since I'm essentially planning to start over, I want to make sure before I do any measurements.

Logged

mwilllems

MC Beta Team
Citizen of the Universe

Posts: 3580



"Linux Merit Badge"
Recipient

Re: Guide to Speaker/Room Correction Using Freeware and JRiver

« Reply #29 on: May 19, 2014, 04:18:49 pm »

Quote from: bluescale on May 19, 2014, 03:17:22 pm

Final question (for now) - Since the goal is to get an idea of the speaker's frequency response, is it fair to assume that I should disable all bass management and other room correction before taking the measurements? I've been operating under that assumption, but it's not explicitly stated in the guide above. Since I'm essentially planning to start over, I want to make sure before I do any measurements.

Short answer: yes, turn off room correction to start. You want to get an idea of what the speaker will do uncorrected, and then work out from there.

You should also initially measure and EQ each speaker separately, meaning that you'd want to disable bass management at first.

Once you've corrected the mains and the sub separately, then you'd need to re-enable bass management (but not necessarily room correction) to see how the mains and sub measure as a composite. Integrating a sub is a lot like designing an active two way speaker meaning it's hard to do correctly without careful measurement.

Once you've gotten each speaker EQed and you're trying to integrate them, we can work all that out in the other thread (as that's an advanced topic not covered in this guide).

Logged

JRiver on Windows 10 64 bit || Arch Linux 64 bit with Gnome
Steinberg UR824 -> Homebrew Bi-Amped Mains ("Project Warhorn")

Check out my Guides:
[Guide to Speaker Correction using JRiver](#)
[Guide to JRiver on the Raspberry Pi](#)
[Casting Audio to JRiver](#)

tyler69

World Citizen

Posts: 160

Re: Guide to Speaker/Room Correction Using Freeware and JRiver

« Reply #30 on: June 15, 2014, 04:12:14 am »

Quote from: mwilllems on February 17, 2014, 12:03:53 pm

3) REW will only generate up to 20 filters, but will generate fewer if it doesn't need all 20. If REW generates 20 filters for your response it probably actually needs more than 20, and you might want to consider generating the filters in two batches (bottom of the range to 600KHz, 601KHz to 10KHz, etc.)

Thank you for this great tutorial mwilllems. I have some questions regarding the filter generation in REW and hope you can help me out:

1. Is it safe to assume, that one can generate filters in more than two batches (i.e. when two batches give me 40 filters)?
2. Is it important that the batches have the same size for both speakers (e.g. 15Hz-500Hz for the left, 15Hz-500Hz for the right?, etc..)?
3. Is it a problem if REW generates more filters for the left speaker, than for the right one?

Logged

mwilllems

MC Beta Team
Citizen of the Universe

Posts: 3580



"Linux Merit Badge"
Recipient

Re: Guide to Speaker/Room Correction Using Freeware and JRiver

« Reply #31 on: June 15, 2014, 09:25:46 am »

Quote from: tyler69 on June 15, 2014, 04:12:14 am

Thank you for this great tutorial mwilllems. I have some questions regarding the filter generation in REW and hope you can help me out:

1. Is it safe to assume, that one can generate filters in more than two batches (i.e. when two batches give me 40 filters)?

Yes that's fine to do, but you might want to confirm that you have your REW target set correctly (and that you've followed other steps above, like making sure FDW is on before export); If you set the target too low you can wind up with many more filters than you otherwise might and will be throwing away quite a lot of gain. There's nothing per se wrong with having a lot of filters, but it's rare (in my experience) to have more than 30 with a correctly set target. Most of the speakers I've seen use that many filters were home-made speakers; I would be surprised to see commercial speakers that needed that much correction.

Quote

2. Is it important that the batches have the same size for both speakers (e.g. 15Hz-500Hz for the left, 15Hz-500Hz for the right?, etc..)?

No, that shouldn't matter as long as the final measurements look ok.

Quote

3. Is it a problem if REW generates more filters for the left speaker, than for the right one?

Not a problem in itself, but depending on how many more it may be a symptom of something.

Every system I've measured had a different number of filters on the left and right, but it's usually a difference of a few filters. Even though speaker manufacturers go to great lengths to "match" speakers, no two speakers will ever actually be identical. So a little difference should be expected.

If you're seeing a difference of more than five or six filters, that means one of two things: either the room is creeping into the measurement in a significant way (which may not necessarily be a bad thing, but is something to keep an eye on) or one of your speakers may be damaged. I once had two titanium cone speakers that measured pretty differently, and then realized that one of them had a dent in the cone :-(

□ Logged

JRiver on Windows 10 64 bit || Arch Linux 64 bit with Gnome
Steinberg UR824 -> Homebrew Bi-Amped Mains ("Project Warhorn")

Check out my Guides:
[Guide to Speaker Correction using JRiver](#)
[Guide to JRiver on the Raspberry Pi](#)
[Casting Audio to JRiver](#)

tyler69

World Citizen

□
□
Posts: 160

□ **Re: Guide to Speaker/Room Correction Using Freeware and JRiver**

« Reply #32 on: June 15, 2014, 03:59:15 pm »

Thank you !

□ Logged

ra7

Recent member

□
Posts: 5

□ **Re: Guide to Speaker/Room Correction Using Freeware and JRiver**

« Reply #33 on: July 05, 2014, 12:19:50 pm »

I have a question regarding Holm and rePhase. I have a three-way system, the details of which can be found here:

<http://www.diyaudio.com/forums/multi-way/258246-corner-expanding-line-array-kef-q100.html>

I'm using miniDSP's USBStreamer for 6 channels of output using JRiver's convolution engine. Using rePhase to develop the filters. Details here:

<http://www.diyaudio.com/forums/pc-based/256809-setting-up-pc-based-multichannel-dsp-system.html>

My question is regarding the delay in piping it through JRiver. There is the delay due to the convolution filters. rePhase gives an indication of this delay for each filter that is generated. And then there is the relative phase due to the physical separation of the drivers. When the measurement is made, I get a different delay in Holm for every single measurement. Even when measuring the same driver consecutively, there is a different sample offset in Holm. Forget about using time lock to get the offset between drivers, there is not even the same offset for the same driver.

I'm using the ASIO loopback in JRiver. The WASAPI loopback doesn't work without stuttering. But I tried it with stutters and it still gave different delays. I've tried all sorts of combination for buffering through JRiver and through the USBStreamer's driver panel. Nothing seems to alleviate the problem of different delays for consecutive measurements.

As a sanity check, I connected the output of the EMU0204 directly to the amp and made consecutive measurements and every time I get the same exact number of sample offsets. So, the problem has to be the loopback through JRiver. Any idea about how this could be solved? I'm guessing the different delays are due to the live processing of the convolution filters in JRiver.

□ Logged

mwillems

MC Beta Team

Citizen of the Universe

□
□
Posts: 3580

□ **Re: Guide to Speaker/Room Correction Using Freeware and JRiver**

« Reply #34 on: July 05, 2014, 01:30:30 pm »

Quote from: ra7 on July 05, 2014, 12:19:50 pm

My question is regarding the delay in piping it through JRiver. There is the delay due to the convolution filters. rePhase gives an indication of this delay for each filter that is generated. And then there is the relative phase due to the physical separation of the drivers. When the measurement is made, I get a different delay in Holm for every single measurement. Even when measuring the same driver consecutively, there is a different sample offset in Holm. Forget about using time lock to get the offset between drivers, there is not even the same offset for the same driver.

I experience the same variable delay from Holm when using loopback, except that when I use the time lock function I get a correct result (i.e. I get a correct inter-channel offset when measuring using the time lock). I think time-lock "holds the stream open" which typically produces consistent results between the two measurements.

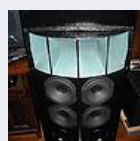
If the time lock doesn't work for you there are a number of alternative methods for measuring interchannel delay. Probably the easiest is the "greatest null method." To accomplish that you'll need: 1) pink noise (JRiver can generate it) and 2) software that can be used as an RTA (REW or TrueRTA, for example); here's how it works:

1) Begin playing pink noise from two channels. For the sake of discussion, we'll assume you're starting with the left tweeter and the left mid-range.

2) Set up your mic at the acoustic center of the two stages and about four feet away. Start your RTA software.

3) Open JRiver's parametric equalizer module, and create a filter to invert the second channel (i.e. the mid-range in this case), and create a second filter to delay the second channel. I recommend initially setting the delay to whatever the physical offset between the drivers would suggest the delay should be, but you can start from zero if you prefer.

4) Now gradually increase the delay on the delay filter and watch the RTA at the crossover point. There will be some cancellation at the crossover point (a null), and you want to keep increasing the delay until you've made the null as deep as it can be: for a while it will keep getting deeper as delay increases, and then will start filling back in. JRiver can adjust the delay in hundredths of a millisecond, so take some time to get the



"Linux Merit Badge" Recipient

delay setting just as deep as possible (it should be at least 25 or 30dB deep and at least an octave across).

5) That delay (the delay that produces the greatest null) is the correct delay. When you de-invert the second channel the two should sum perfectly at the crossover (assuming you've chosen crossover slopes that sum flat).

You would then want to repeat the process to sync your mid-range and woofer, etc. (for it to work, the two stages being measured must share a crossover point, obviously).

The nice thing about the greatest null method is that it is completely immune to variable front-end latency because it only measures the actual offset between the channels as they are playing (including all sources of delay).

If you're trying to use the time lock to do something other than set interchannel delay, let me know.

Hope this helps!

▢ Logged

JRiver on Windows 10 64 bit || Arch Linux 64 bit with Gnome
Steinberg UR824 -> Homebrew Bi-Amped Mains ("Project Warhorn")

Check out my Guides:
[Guide to Speaker Correction using JRiver](#)
[Guide to JRiver on the Raspberry Pi](#)
[Casting Audio to JRiver](#)

ra7

Recent member

▢
Posts: 5

▢ **Re: Guide to Speaker/Room Correction Using Freeware and JRiver**

« Reply #35 on: July 05, 2014, 01:46:28 pm »

Thanks for the prompt reply!

The greatest null method can be useful. I did something similar in Holm to arrive at the approximate delay. What you're banking on though, is that you have perfect phase overlap between the drivers after the crossover. When you're starting the design, you just don't know that. There are many things that affect the relative phase between drivers: physical separation, crossover slopes, driver amplitude response. To this must be added the delay due to the convolution filter for each driver. Now, we're adding another variable, the delay in processing in JRiver, that cannot be precisely accounted for. To summarize, you can get close to the right delay, but there is no way to characterize the relative phase between the drivers.

Are you sure time lock works for you? I tried it and it gives different offsets for every measurement. For example, two separate measurements of tweeter+time lock+woofer give different offsets.

A sure shot way to do this is not use JRiver at all. Instead, import the convolution filters in Holm, make a direct measurement of the drivers and then convolve the measured response with the filter in Holm. But this would be very tedious to do during crossover design. It is much easier to do measurements and tweaking on the fly.

Is there no way to buffer the outputs or do something so that the delay through JRiver is fixed?

▢ Logged

mwillems

MC Beta Team
Citizen of the Universe

▢▢▢▢
Posts: 3580



"Linux Merit Badge"
Recipient

▢ **Re: Guide to Speaker/Room Correction Using Freeware and JRiver**

« Reply #36 on: July 05, 2014, 02:17:19 pm »

Quote from: ra7 on July 05, 2014, 01:46:28 pm

The greatest null method can be useful. I did something similar in Holm to arrive at the approximate delay. What you're banking on though, is that you have perfect phase overlap between the drivers after the crossover. When you're starting the design, you just don't know that. There are many things that affect the relative phase between drivers: physical separation, crossover slopes, driver amplitude response. To this must be added the delay due to the convolution filter for each driver. Now, we're adding another variable, the delay in processing in JRiver, that cannot be precisely accounted for. To summarize, you can get close to the right delay, but there is no way to characterize the relative phase between the drivers.

I think you may be working from a faulty assumption, or I may be misunderstanding you (in which case, sorry for being dense).

JRiver's loopback does not (in any measurement I've taken) introduce variable *inter-channel* delay. The total delay of the system due to loopback may vary, but all channels have the same "processing" delay due to loopback in any given playback event. It's just that Holm tends to drop the stream in between measurements unless timelock is engaged. So the delay introduced by JRiver's loopback is irrelevant to normal playback because it is the same for all channels at any given point in time (if that makes sense).

If your specific convolution filters introduce differing amounts of delay, that delay would obviously vary by channel, but would be a fixed delay, which would be captured by the greatest null method. JRiver's convolution engine does not introduce *variable* interchannel delay, otherwise the convolution engine would never work correctly at all.

I can confirm from my own repeated measurements of a convolved bi-amped system using loopback that JRiver does not (at any stage) introduce variable inter-channel delay, and that the greatest null method will produce correct results (subject to the caveats you note).

As for not knowing some things at the start of the design, in my own experience designing an infinitely customizable active crossover can be a pretty fluid process, that may never completely end ☐

I agree that if one has asymmetrical crossovers, or hasn't first corrected the response of one's drivers, one may not get optimal results from the greatest null method on the first try; but one can do those things in any order with an active crossover. I.e. my workflow is to flatten the FR/phase of the channels to half an octave past the crossover point, set the desired crossover slopes, and then set the delay. Done in that order the greatest null usually gives me the correct result on the first try.

Quote

Are you sure time lock works for you? I tried it and it gives different offsets for every measurement. For example, two separate measurements of tweeter+time lock+woofer give different offsets.

Quite sure, I have gotten repeatable measurements that also happen to produce "correct" phase measurements once dialed in. Time-lock also gave me the same delay (within a narrow margin) as the greatest null method. I'm also fairly sure it's working for a few other folks here on the forum as well, maybe they can weigh in on anything special they had to do?

It's been a little while since I used that functionality, and I was using WASAPI loopback at the time, so maybe that's the difference?

Quote from: ra7 on July 05, 2014, 01:46:28 pm

Is there no way to buffer the outputs or do something so that the delay through JRiver is fixed?

Unfortunately, the only way I know of is to use time-lock in Holm (and make sure that the "keep stream open" checkbox is checked in the Holm options). Other than that, I'm not aware of a way to fix the latency.

If you have a spare I/O channel you may be able to use REW to get a robust real-time latency measurement using a literal loopback timing reference, but some folks around here have had a hard time with that.

Sorry I'm not more help on that front.

▢ Logged

JRiver on Windows 10 64 bit || Arch Linux 64 bit with Gnome
Steinberg UR824 -> Homebrew Bi-Amped Mains ("Project Warhorn")

Check out my Guides:
[Guide to Speaker Correction using JRiver](#)
[Guide to JRiver on the Raspberry Pi](#)
[Casting Audio to JRiver](#)

ra7

Recent member

Posts: 5

▢ **Re: Guide to Speaker/Room Correction Using Freeware and JRiver**

▢ « Reply #37 on: July 05, 2014, 04:30:32 pm »

Quote from: mwillems on July 05, 2014, 02:17:19 pm

JRiver's loopback does not (in any measurement I've taken) introduce variable *inter-channel* delay. The total delay of the system due to loopback may vary, but all channels have the same "processing" delay due to loopback in any given playback event. It's just that Holm tends to drop the stream in between measurements unless timelock is engaged. So the delay introduced by JRiver's loopback is irrelevant to normal playback because it is the same for all channels at any given point in time (if that makes sense).

I think we are talking about slightly different things. Obviously convolution wouldn't work if all channels didn't have the same delay. However, that delay changes from one moment to another. If you measure one driver at 2:00 pm, the total delay out of JRiver may be 500 ms. When you measure the same driver again at 2:01 pm, the total delay is not necessarily 500 ms. This is what trips up the measurement in Holm.

Quote

Quite sure, I have gotten repeatable measurements that also happen to produce "correct" phase measurements once dialed in. Time-lock also gave me the same delay (within a narrow margin) as the greatest null method. I'm also fairly sure it's working for a few other folks here on the forum as well, maybe they can weigh in on anything special they had to do?

Time lock is nothing but using the same offset for the second driver as the first (AFAIK). You can set that offset manually as well in the options box. Yes, the 'Keep in/out stream active' checkbox must be checked. But for the offset to work, the two measurements should have the same delay out of JRiver.

Quote

If you have a spare I/O channel you may be able to use REW to get a robust real-time latency measurement using a literal loopback timing reference, but some folks around here have had a hard time with that.

I may have to try that sometime.

Having said all that, I think the issue is solved by using WASAPI instead of ASIO for the USBStreamer output in JRiver. Now for consecutive measurements of the same driver, I get a delay of 5 samples ($5/44.1 = 0.11$ ms... not too bad). Each consecutive measurement adds 5 samples. Strange, but predictable. This is good enough for doing crossover work.

Thanks a lot for your insight and help! Greatly appreciated.

▢ Logged

mwillems

MC Beta Team
Citizen of the Universe

Posts: 3580



"Linux Merit Badge"
Recipient

▢ **Re: Guide to Speaker/Room Correction Using Freeware and JRiver**

▢ « Reply #38 on: July 05, 2014, 06:57:32 pm »

Quote from: ra7 on July 05, 2014, 04:30:32 pm

Having said all that, I think the issue is solved by using WASAPI instead of ASIO for the USBStreamer output in JRiver. Now for consecutive measurements of the same driver, I get a delay of 5 samples ($5/44.1 = 0.11$ ms... not too bad). Each consecutive measurement adds 5 samples. Strange, but predictable. This is good enough for doing crossover work.

Thanks a lot for your insight and help! Greatly appreciated.

Great news! Glad you got it working ☐


▢ Logged

JRiver on Windows 10 64 bit || Arch Linux 64 bit with Gnome
Steinberg UR824 -> Homebrew Bi-Amped Mains ("Project Warhorn")

Check out my Guides:
[Guide to Speaker Correction using JRiver](#)
[Guide to JRiver on the Raspberry Pi](#)
[Casting Audio to JRiver](#)

Denti

▢ **Re: Guide to Speaker/Room Correction Using Freeware and JRiver**

Galactic Citizen <div> <div></div> <div></div> <div></div> <div></div> </div> Posts: 397	<div>« Reply #39 on: July 08, 2014, 05:15:08 pm »</div> <div>I'd like to try this but need recommendations for a cheap but effective mic.</div> <div> <div></div> <div>Logged</div> </div>
natehansen66 World Citizen <div> <div></div> <div></div> <div></div> <div></div> </div> Posts: 238	<div> <div>Re: Re: Guide to Speaker/Room Correction Using Freeware and JRiver</div> <div>« Reply #40 on: July 09, 2014, 04:59:45 am »</div> </div> <div>I use the dual channel method in REW with Wasapi loopback and don't have any issues. That said I've never tried it with convolution.</div> <div>Denti - the cheapest mic would be a USB mic. Personally I don't think it's optimal because you can't do dual channel measurements like I was speaking of but it's cheaper than a mic + preamp. I would spend a bit more and get a professionally calibrated unit from Cross Spectrum Labs.</div> <div> <div></div> <div>Logged</div> </div>
mwillems MC Beta Team Citizen of the Universe <div> <div></div> <div></div> <div></div> <div></div> </div> Posts: 3580 <div>  <div>"Linux Merit Badge" Recipient</div> </div>	<div> <div>Re: Guide to Speaker/Room Correction Using Freeware and JRiver</div> <div>« Reply #41 on: July 09, 2014, 08:57:08 pm »</div> </div> <div>Quote from: Denti on July 08, 2014, 05:15:08 pm</div> <div>I'd like to try this but need recommendations for a cheap but effective mic.</div> <div>I'll second the recommendation for a calibrated phantom-powered mic from cross-spectrum if you have a mic pre-amp handy. The calibrated USB mics that Parts Express sells are not bad either if you're looking for something cheaper.</div> <div>USB mics do have some disadvantages, but they're cheaper and easier if you don't have a mic pre-amp on hand.</div> <div> <div></div> <div>Logged</div> </div> <div>JRiver on Windows 10 64 bit Arch Linux 64 bit with Gnome Steinberg UR824 -> Homebrew Bi-Amped Mains ("Project Warhorn")</div> <div>Check out my Guides: Guide to Speaker Correction using JRiver Guide to JRiver on the Raspberry Pi Casting Audio to JRiver </div>
ra7 Recent member <div> <div></div> <div></div> <div></div> <div></div> </div> Posts: 5	<div> <div>Re: Guide to Speaker/Room Correction Using Freeware and JRiver</div> <div>« Reply #42 on: July 16, 2014, 05:18:48 pm »</div> </div> <div>Using the dual measurement system is a good idea. But it will be a little painful to keep changing the second channel measurement input, either out of the amplifier or the DAC, as you measure each of the different channels.</div> <div>What would be ideal is if somehow JRiver can keep the same delay out of two consecutive measurements with convolution enabled.. call it measurement mode with convolution.</div> <div> <div></div> <div>Logged</div> </div>
natehansen66 World Citizen <div> <div></div> <div></div> <div></div> <div></div> </div> Posts: 238	<div> <div>Re: Guide to Speaker/Room Correction Using Freeware and JRiver</div> <div>« Reply #43 on: July 26, 2014, 11:20:55 am »</div> </div> <div>Why would you have to change the second channel (if I understand you correctly you're talking about the loopback reference channel)? Once I have it setup I don't have to change any physical connections. I set MC to output an un-filtered signal to an unused channel on my DAC (granted I have 12 channels with 4 not used), and that goes to the second input on my mic preamp. I've never had any issues doing it this way. Looking back at your thread on DIYA it looks like you have 8 output channels on your DAC so you should be able to work with one speaker at a time, then switch to do the second speaker. Is that what you mean?</div> <div> <div></div> <div>Logged</div> </div>
ra7 Recent member <div> <div></div> <div></div> <div></div> <div></div> </div> Posts: 5	<div> <div>Re: Guide to Speaker/Room Correction Using Freeware and JRiver</div> <div>« Reply #44 on: July 28, 2014, 05:51:03 pm »</div> </div> <div>Yes, I do have some spare channels and I could use them for the measurement. But as we have seen, latency is variable in loopback mode. So, say, after measuring one driver, when I have to measure the second driver, I cannot get the convolved output on the same channel as before, right? I'd have to move the probe to a different channel.</div> <div>For the latest iteration of the crossover, I did everything outside JRiver. Measured raw response (EMU -> amp -> speaker), then exported response to rePhase, exported filters from rePhase and convolved raw response with filters in Holm. Then, with everything put together and loopback in JRiver, I measured the total response and compared it to the summed response in Holm. Got pretty close.</div> <div> <div></div> <div>Logged</div> </div>
packux Recent member <div> <div></div> <div></div> <div></div> <div></div> </div> Posts: 43	<div> <div>Re: Guide to Speaker/Room Correction Using Freeware and JRiver</div> <div>« Reply #45 on: December 27, 2014, 06:37:25 am »</div> </div> <div>Hi all and Happy New Year,</div> <div>I tried to follow this how-to and it seems I have some problems.</div>

My current set-up is as following:

1. DAC/ADC: Metric Halo Lio-8
2. PC interface: Lynx AES16e
3. I am currently using Lynx inputs/outputs 0-8.
4. I have set-up input 3 from Lio to be directly routed to Digital In 3
5. I am using an ECM 8000 with a Rols PB23
6. JRiver is configured to use the ASIO Lynx driver.

Using either Holms (with DirectSound) or REW (with Java) and setting as input Lynx Recording 03+04 (WDM driver) and as output JRiver WDM has any of the following effects

1. No sound output from JRiver. It seems as if no audio goes through JRiver's engine as the spectrometer in JRiver is showing no audio activity. Holms (or REW) seem to be receiving audio from the digital in
2. No input either to REW or Holms. However, Audio output from JRiver, either Pink noise from JRiver itself (using Tools > Advanced Tools > Audio calibration in JRiver) or noise from measurement software seems to be working.

When using Holms or REW with ASIO, I can perform measurements, however it is not possible for me to measure the effect of corrections performed in JRiver DSP.

I remember reading multiple times that the Lynx ASIO driver does not play well with multi-use scenarios. As it currently stands, what I am trying to do is to use ASIO output (JRiver) and DirectAudio Input (WDM driver provided by Lynx) which I would guess is such a scenario.

One idea I had was to use JackRouter and JRiver but before going down that path, is there anything else that I could do which would not introduce so much latency/complexity?

Thanks already for your time everybody.

□ Logged

mwillems

MC Beta Team
Citizen of the Universe



Posts: 3580



"Linux Merit Badge"
Recipient

□ **Re: Guide to Speaker/Room Correction Using Freeware and JRiver**

« Reply #46 on: December 27, 2014, 07:06:34 am »

Quote from: packux on December 27, 2014, 06:37:25 am

Hi all and Happy New Year,

I tried to follow this how-to and it seems I have some problems.

My current set-up is as following:

1. DAC/ADC: Metric Halo Lio-8
2. PC interface: Lynx AES16e
3. I am currently using Lynx inputs/outputs 0-8.
4. I have set-up input 3 from Lio to be directly routed to Digital In 3
5. I am using an ECM 8000 with a Rols PB23
6. JRiver is configured to use the ASIO Lynx driver.

Using either Holms (with DirectSound) or REW (with Java) and setting as input Lynx Recording 03+04 (WDM driver) and as output JRiver WDM has any of the following effects

1. No sound output from JRiver. It seems as if no audio goes through JRiver's engine as the spectrometer in JRiver is showing no audio activity. Holms (or REW) seem to be receiving audio from the digital in
2. No input either to REW or Holms. However, Audio output from JRiver, either Pink noise from JRiver itself (using Tools > Advanced Tools > Audio calibration in JRiver) or noise from measurement software seems to be working.

When using Holms or REW with ASIO, I can perform measurements, however it is not possible for me to measure the effect of corrections performed in JRiver DSP.

I remember reading multiple times that the Lynx ASIO driver does not play well with multi-use scenarios. As it currently stands, what I am trying to do is to use ASIO output (JRiver) and DirectAudio Input (WDM driver provided by Lynx) which I would guess is such a scenario.

One idea I had was to use JackRouter and JRiver but before going down that path, is there anything else that I could do which would not introduce so much latency/complexity?

Thanks already for your time everybody.

Forum user mojave has some Lynx hardware, and as I recall found a multi-client workaround using the a Steinberg multiclient ASIO patch/driver? I'm on my phone so I can't find the exact posts where he discusses it, but there was definitely a thread about it a while back. If he hasn't posted by the time I'm back in front of a keyboard I'll try and hunt it up.

EDIT: I found a link to the driver. To be clear, folks on the forum have used this driver for a variety of ASIO interfaces to include RME and Lynx interfaces. YMMV, but here it is:

<http://www.steinberg.net/forums/viewtopic.php?f=34&t=48>

□ Logged

JRiver on Windows 10 64 bit || Arch Linux 64 bit with Gnome
Steinberg UR824 -> Homebrew Bi-Amped Mains ("Project Warhom")

Check out my Guides:

[Guide to Speaker Correction using JRiver](#)

[Guide to JRiver on the Raspberry Pi](#)

[Casting Audio to JRiver](#)

mojave

MC Beta Team
Citizen of the Universe

□ **Re: Guide to Speaker/Room Correction Using Freeware and JRiver**

« Reply #47 on: December 27, 2014, 07:47:10 am »

Posts: 3359

Requires "iTunes or better" so I installed JRiver

Quote from: mwillems on December 27, 2014, 07:06:34 am

If he hasn't posted by the time I'm back in front of a keyboard I'll try and hunt it up.

You sure didn't give me much time.

Logged

packux

Recent member

Posts: 43

Re: Guide to Speaker/Room Correction Using Freeware and JRiver

« Reply #48 on: December 27, 2014, 12:40:14 pm »

Thanks both for the update.

Yes I indeed found this workaround myself while trying to figure out a way to move forward but It didn't do much.

In practice, I have managed to measure (i.e. get input and output), however up until now it has not been possible to pass audio through JRiver audio engine.

I also have tried asio4all with the intention to create a layer on top of JRiver / lynx where one devices is used for input and another for output.

That's where I started thinking Jack could help out. A scenario I think of trying is the following.

1.open asio in via JRiver in a specific zone and output to Jack.

2.read from Jack in Holm

3.output from Holm to Jack

4.jack to JRiver ASIO

ideas?opinions?

Logged

packux

Recent member

Posts: 43

Re: Guide to Speaker/Room Correction Using Freeware and JRiver

« Reply #49 on: December 28, 2014, 04:15:22 am »

I just played a bit around the idea I was proposing, but it doesn't seem to work in any way.

1. Using 'Open Live... > ASIO Line In' in JRiver doesn't work for me. In ASIO Control panel, I configured it to open ASIO Lynx with 1 channel and channel offset 1 but I got the dreaded message 'Something went wrong with the playback'. FYI, the sampling rate available when configuring the ASIO Line in is also not what I would expect, as it provides sampling rate up to 96Khz (I have sampling rate configured to 192Khz in MH and Lynx)

2. It is not very clear how to connect REW and JRiver. I would use the ASIO Line in of Jack and use Jack as input/output in REW, but there I am stuck a bit from Jack configuration capabilities. I will most likely have to further check how things work there.

Any ideas on (1)?

Logged

Pages: [1] 2 3 Go Up

« previous next »

Jump to: => Sound Cards, DAC's, Receivers, Speakers, and Headphones

go

Page created in 0.114 seconds with 19 queries.

About Media Center: [Forum](#) | [Getting Started](#) | [Purchase](#)
Support: [Forum](#) | [Wiki](#) | [Developer](#) | [Licensing](#) | [Privacy Policy](#) | [Links](#)
Corporate Solutions: [Overview](#) | [Customers](#) | [Company](#)
Other Products: [ICE](#) | [Media Center Remote](#)
Copyright © 2010-2016 JRiver, Inc. All Rights Reserved.
Powered by [SMF 1.1.21](#) | [SMF](#) © 2015, Simple Machines